AD

R-TR-77-025

A DYNAMIC ANALYSIS OF THE SELF-PROPELLED, 8-INCH, M110A1, HOWITZER UTILIZING FORMAC

THOMAS D. STREETER
ROBERT H. COBERLY

JUNE 1977

TECHNICAL REPORT

LARGE CALIBER WEAPONS SYSTEMS LABORATORY

US ARMY ARMAMENT RESEARCH
AND DEVELOPMENT COMMAND
ROCK ISLAND, ILLINOIS 61201

Streeter 5292

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
R-TR-77-025

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
A Dynamic Analysis of the Self-Propelled 8" M110A1 Howitzer Utilizing FORMAC

5. TYPE OF REPORT & PERIOD COVERED
FINAL REPORT
Sep 76 - Apr 77

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
THOMAS D. STREETER
Robert H. COBERLY

8. CONTRACT OR GRANT NUMBER(s)
N/A

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Weapons Division
Large Caliber Weapons System Laboratory
Dover, NJ 07801

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
6.26.03.A
1W662603AH78

11. CONTROLLING OFFICE NAME AND ADDRESS
Weapons Division
Large Caliber Weapons Systems Laboratory
Dover, NJ 07801

12. REPORT DATE
June 1977

13. NUMBER OF PAGES
150

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)
Unlimited

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

M110A1	SIMULATION
FORMAC	FORTRAN
DYNAMIC	SELF-PROPELLED
MATH MODEL	LAGRANGE

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report presents a new method for solving Lagrange's equations of motion utilizing FORMAC. An application using this technique is given with an eleven degree-of-freedom problem which describes the motion of the M110A1, a self-propelled 8" howitzer under dynamic conditions of firing. A computer program has been written, is operational, and the listing is contained in the appendix. This report is an endeavor to automate the generation and solution of the equations of motion for dynamical systems.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

CONTENTS

ABSTRACT

This report presents a new method for solving Lagrange's equations of motion utilizing FORMAC. An application using this technique is given with an eleven degree-of-freedom problem which describes the motion of the M110A1, a self-propelled 8" howitzer under dynamic conditions of firing. A computer program has been written, is operational, and the listing is contained in the appendix. This report is an endeavor to automate the generation and solution of the equations of motion for dynamical systems.

FORWARD

In October 1976, it was requested that a mathematical model be developed of the M110A1 to study the dynamic motion of the system during firing so as to define dynamic loading at various points in the weapon system for a subsequent stress analysis of weapon components. This project was funded by AMC under AMCMS Code 664602.12.38900, D. A. Project No. 1W664602D389.

The authors are indebted to Miss Ann Marie Lowder and Mrs. Jan Keller for their many hours of typing in preparing this report. Also, the authors appreciate the assistance given by Mr. Al Smith in helping to obtain the data for this model.

1.0 INTRODUCTION

1.1 General Discussion of Modeling

A convenient way to describe a physical system and study its behavior is to represent the system by mathematical equations. Generally, the resulting equations are a set of differential equations based on the laws of mechanics and the geometry of the physical system. The use of a computer is usually necessary to solve this set of equations and especially if the system under study is complex and a sufficient amount of detail is required to characterize it. If the use of a computer becomes necessary, the mathematical expressions, physical constraints and logic must be transformed into a computer program. The equations and their solution technique comprise a mathematical or computer model. If such a model is to be of value, this model must be amenable to analysis and its output must represent the behavior of the physical system with sufficient accuracy so that useful information can be obtained concerning the actual system.

Since real world conditions cannot be modeled exactly, inherent inaccuracies will exist in the model because the modeler must resort to simplifications and abstractions of the actual, physical system. The important point is that an acceptable level of confidence be established so that inferences drawn from the model output are correct and the model generates the same behavior characteristics as the actual system. Confidence in the model output is gained through model validation.

Model validation is a check of the agreement between the behavior of the model and that of the actual system. The correctness of the model can only be measured relative to the physical system. Models may be validated according to various criteria. One such criteria is Bayes' formula for conditional probabiliities. With this procedure, specific model outputs are compared with field test data, and the conditional probability (that the model is valid given the field test results) is determined.

However, many times the system to be modeled is still on the drawing board and field test results do not exist. If this is the case, the design and simulation should complement each other. That is, a math model may be developed while the design of the system is progressing from the drawing board stage to system integration. Here the model may be used to provide insight into how the system will perform under dynamic conditions. Thus the model may be used to (1) predict dynamic loading on critical parts, (2) define sensitivity of weapon performance to design parameters, (3) evaluate suggested weapon modifications and (4) provide the necessary foundation for design optimization studies.

As test data becomes available, it is necessary to "validate" and "tune" the model. If a correlation between test results and predicted motion is poor and significant motion occurs which was neglected in the model, a complete revision of the model may be required. However, if all significant motion has been accounted for in the model but the magnitude and timing of the predicted motion is in error, a "tuning" of the model is in order. For example, the model to be generated in this report requires an effective spring rate for the ground-weapon interaction. The value chosen for this parameter may be poorly estimated. If varying the value of this parameter between reasonable limits greatly improves the desired correlation between model and test, the model is said to have been tuned.

In the design, development, and fabrication of any given system, the proper mixture of analysis, simulation, and testing is necessary to produce the payoffs that must be achieved for a cost-effective product. The use of model studies may reduce costs in lieu of extensive testing, but it does not necessarily preclude the need for performing an actual field test. A field test yields information for the evaluation of the system itself, provides a data base to be used for comparison to other systems, and is necessary for model validation.

One of the most important and useful tools to analyze the results of a given design is through mathematical simulation. Such is the case for the development of the model contained in this report. Simulation is a powerful tool and it enables the analyst to become familiar with the behavior or performance of the actual system under study when subjected to a variety of different conditions or parameter changes. Generally, experiments with the actual system itself are very costly; however, they can be performed on the model with relative ease and at low cost. Many times, experimentation on the model will provide more information about the interaction of variables than testing on the actual system because of the controlled environment and the ease of parameter variation.

Weapon systems of today are becoming more complex and this trend is likely to continue because of the current threats being proposed; as a result, design requirements are becoming increasingly more difficult to satisfy. It is well recognized that a change in design or a change in even one aspect of the weapon system may very well produce changes or create the need for changes in other parts of the system. As the long list of parameter trade-offs such as accuracy, caliber, threat, dispersion, etc., are considered, the engineer's intuition and experience become increasingly more difficult to apply and it becomes more important to define the design procedure mathematically. Design requirements specify that a weapon system is to perform some task at some index of performance. Thus, the design of a weapon system provides a natural setting for an optimization problem, assuming a knowledge of all environmental factors influencing the design process are known. To just search for admissible parameters so that the system is enabled to perform its task is not satisfactory. It should be required to seek those parameters so that performance is optimized (in some sense).

In the past, conventional methods of analyzing the dynamic behavior of a given system required the analyst to linearize the generalized coordinates to achieve simplification of expressions. This is no longer necessary. FORMAC, a language developed by IBM, allows the analyst to proceed with a nonlinear analysis of the system being studied. It also allows mathematical models suitable for optimization studies to be formulated with relative ease (as far as obtaining the required differential expressions).

The model developed in this report is completely nonlinear. The reasons for this approach and an in-depth discussion of its derivation are given in the sections that follow.

1.2 Description of the Model

The purpose of this technical report is to document the work accomplished to date on the development of a mathematical model for the Howitzer, Heavy, Self-Propelled, Full-Tracked, 8-inch, M110A1, see figure 1.1. This documentation has two major objectives; (1) the detailed description of a new method for developing and solving Lagrange's equation of motion and (2) the generation of a mathematical simulation to describe the motion of the system and to define dynamic loading at various points in the weapon (which are to be used as input for a stress analysis of weapon components).

A method was developed, utilizing FORMAC, to obtain the necessary symbolic representations for the differential expressions required to solve Lagrange's equations of motion of the M110A1. FORMAC, as developed by IBM, provides for the symbolic manipulation of mathematical expressions, i.e., the expression SIN(X) can be differentiated, resulting in the expression COS(X). Expressions can be differentiated, evaluated, replaced, compared, and parsed. After differentiation, expressions which occur repeatedly can be replaced by new variable names and thus millions of arithmetic operations can be eliminated during the execution of the computer program. This is illustrated in appendix E. Since PL/I is a subset of FORMAC, all of the facilities

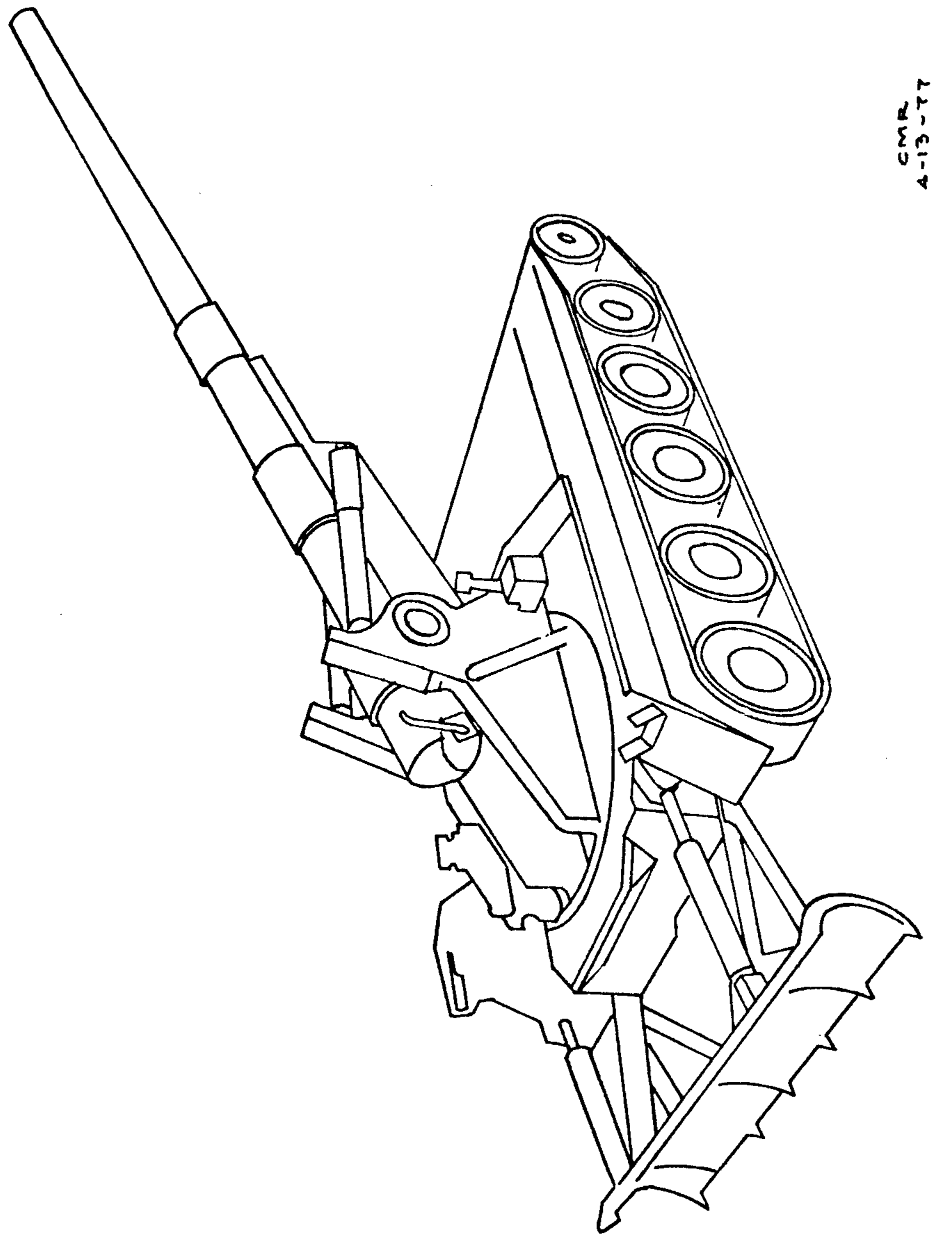


Figure 1.1

of PL/I are available for program structure, loop control and I/O. The FORMAC output consists of coded differential expressions and they are automatically punched on cards (error free) in FORTRAN format. The end product is a computer program written in FORTRAN. Approximately 95 percent of the work to generate the simulation model was accomplished on the computer. This procedure essentially reduced the 11 degree-of-freedom problem described below to a rather routine operation.

In the development of mathematical models of weapon systems it is desirable to limit the degrees of freedom of the model to the major gross motions of the system. In general, the larger the model (in terms of degrees of freedom) the less accurate is the predicted motion of the system. This is primarily due to the difficulty in accurately defining the values of the weapon parameters. However, this problem can be overcome by performing carefully instrumented tests (both static and dynamic). If a physical system does exhibit significant multi-degree of freedom motion it is necessary to model that motion and accept the complexity of the resulting system of equations. In this report it was essential that all major motions be identified, resulting in an eleven degree-of-freedom system.

The model developed for the M110A1 has five distinct masses: (1) the vehicle, M_V, (2) the spade assembly, M_S, (3) those parts that traverse but do not elevate, M_T, (4) those parts that elevate but do not recoil, M_E, and (5) the recoiling parts, M_R.

The generalized coordinates were defined based on the following logic: the entire system would translate laterally (x), roll (θ), and yaw (ψ) as a single rigid mass; the vehicle would translate fore and aft (y), translate in a vertical direction (z), and pitch (ϕ) about its own mass center; those parts that traverse ($M_T + M_E + M_R$) would yaw (τ) relative to the vehicle; those parts that elevate ($M_E + M_R$) would pitch (γ) relative to the vehicle; the recoiling parts would translate (η) relative to the M_E parts; finally the spade assembly would translate fore and aft (v) and pitch (ν) relative to the

ground. Note the initial value of γ (γ_0) is the angle of elevation, the initial value of τ (τ_0) is the angle of traverse, and the initial value of η (η_0) locates the in battery position of the mass center of the recoiling parts relative to the trunnion.

A brief description of the operation of the physical system is given below. When the weapon is fired, the pressure generated by the burning propellant drives the projectile out of the tube and forces the recoiling parts (tube, recoil rods, recoil pistons, etc.) rearward. The motion of the recoiling parts is primarily resisted by oil pressure on the face of the recoil piston. The piston is pulled through a cylinder of oil, the oil being throttled around the piston via orifice areas which are designed as a function of recoil displacement to minimize peak recoil forces. In parallel, to the recoil cylinder is the recuperator cylinder wherein the recuperator piston, during recoil, compresses a gas and hence stores enough energy to return the recoiling parts to the firing (in-battery) position. The total recoil force is transferred to the understructure through the trunnions. As a result of this system of forces, a large torque is generated around the trunnions and the elevating parts will tend to pitch relative to the vehicle. The restraint to this pitching motion is offered by a friction brake. Until the moment tending to produce pitch exceeds the restraining moment of the friction brake, no pitch motion results. When the moment tending to produce pitch exceeds the restraining moment, pitch motion is initiated; and will continue until the constant restraining moment of the brake brings the pitching motion to a stop. A similar action occurs in the yaw motion of the traversing parts.

The spades are designed to offer ground resistance to rearward motion and to the pitching of the spade assembly but little resistance to roll, yaw and translations in the lateral and vertical directions. Thus horizontal ground springs acting on the spade assembly in the longitudinal direction are sufficient to restrain the longitudinal translation and pitch of the spade assembly.

The resistance to lateral translation and yaw of the weapon system is the ground friction between the ground and the vehicle tread (assumed to be at the ground contact point of the four corner roller wheels). Vertical ground springs are located at these four ground contact points and restrain the roll and vertical translation of the weapon system. Note the effective spring rate goes to zero at any of the ground contact points when contact between the ground and the vehicle is lost. The braces and spade cylinders are modeled as springs and hydraulic damping and act primarily in the longitudinal direction between the vehicle and the spade assembly. Dampers are associated with the vertical ground springs and the braces.

The method of mathematically describing the above physical phenomina is contained in Appendix A.

2.0 SOLUTION TECHNIQUE

2.1 Overview

Experience has demonstrated that modeling multi-degree of freedom systems requires many hours of tedious manipulation of expressions with the risk of generating numerous errors. The result is generally the implementation of a model which is a linearized version of the system under consideration. That is, in the conventional technique, the analyst is forced to linearize most of the generalized coordinates to achieve simplification of expressions. These shortcomings led to the development of a semi-automated procedure.

This report develops a new technique for obtaining the dynamic equations of motion of any system resulting from Lagrange's equation. The equations generated by the procedure given here are completely nonlinear. This approach is taken for several reasons. First, the accuracy that might be lost by linearizing due to cross-coupling effects is not known; secondly, the nonlinear model is much easier to obtain than the linearized version using the methods developed in this report and the nonlinear approach does not require much, if any, additional core storage; and thirdly, a more accurate model would produce results which are closer in agreement to that of the real world.

It was found that the term $-\partial$(Kinetic Energy)/∂(Generalized Coordinates) does not have to be calculated as the positive of this term appears in the expression $\frac{d}{dt}$ ∂(Kinetic Energy)/∂(Generalized Velocities) and they cancel. This result is always the case. If the procedure in this report is used, two, three, and perhaps even four degree-of-freedom systems can be accomplished quite easily by hand if the FORMAC software package is not available to the analyst.

2.2 Lagrange's Equation

The expression for the Lagrangian method which yields the equations of motion for a dynamical system is

$$\frac{d}{dt}\frac{\partial(KE)}{\partial\dot{q}_j} - \frac{\partial(KE)}{\partial q_j} + \frac{\partial(DE)}{\partial\dot{q}_j} + \frac{\partial(PE)}{\partial q_j} = F_j \qquad (2.1)$$

where $j = 1,2,\ldots,k$

KE = Total kinetic energy

DE = Total dissipative energy

PE = Total potential energy

F_j = Generalized external force

q_j = Generalized coordinate

$\dot{q}_j$ = Generalized velocity

t = Independent variable, time

k = Number of generalized coordinates

Equation (2.1) in matrix form can be written as

$$A(q)\ddot{q} = B(q,\dot{q},t) \qquad (2.2)$$

The first term of equation (2.1) will generate the A matrix plus additional terms which contribute to the B vector. The remaining four terms of equation (2.1) make up the rest of B. It will be seen in the sequel that equation (2.1) can be expressed conveniently in matrix form.

In generating the desired equations of motion by utilizing FORMAC, it becomes necessary to examine Lagrange's equation in detail, see Appendix B. This is required because FORMAC performs only partial differentiation. The results of the analysis in the appendix determine which derivatives are to be taken and how they are ultimately combined to yield a set of 2nd order nonlinear differential equations describing the dynamic motion of the system under investigation.

In the conventional method of actually carrying out the analysis of the Lagrangian, an imbedding of terms occurs within a given expression (terms occur repeatedly) due to the matrix operations, dot products, and the way in which each succeeding mass center is located from the preceeding one. To avoid the imbedding problem as much as possible, the philosophy of the technique reported here is to operate

on the position vectors of each mass before dot products are actually taken. In fact, because of the way each succeeding mass center is located, the position vectors themselves can be broken up into smaller expressions.

However, imbedding still occurs to some extent, but the expressions are small enough such that the imbedded terms can be identified rapidly and replaced by new variable names quite easily with FORMAC. Thus, millions of arithmetic operations are saved in the course of actually solving the differential equations during the running of the computer program. To see this, the reader should refer to Appendix E to get some idea of how much of a reduction in the number of arithmetic operations can actually be accomplished by the replacement of repeated terms with new variable names. All of the partial differentiation as required by Appendix B is performed on the smaller expressions and then a reduction in the size of these differential expressions is obtained by removing the imbeddedness. The dot product of the position vectors is taken at a later time with numbers as elements instead of with large expressions as is normally done in the conventional method. FORMAC necessitated the analysis covered in Appendix B since only partial derivatives could be obtained by using FORMAC. The outcome of this analysis provides a technique that facilitates the derivation of the equations of motion whether done by hand or with the aid of a computer.

2.3 FORMAC Approach

An objective of the work reported here was to develop an organized and efficient computational scheme that would handle multi-degree of freedom problems with relative ease. FORMAC provides this capability. It is an IBM software package which allows the manipulation of mathematical expressions. Using FORMAC, expressions can be differentiated, evaluated, replaced, compared, and parsed. Since PL/I is a subset of FORMAC, all of the facilities of PL/I are available for program structure, loop control and I/O. The computer performs the necessary matrix

multiplication and differentiation. In addition, angles are coded, expressions are optimized and finally punched in FORTRAN format ready for numerical integration. This process virtually eliminates any mathematical or key punching errors for approximately 85 percent of the FORTRAN program as this amount of the computer program is obtained from the results of the FORMAC output. This technique also allows the analyst to spend considerably more time formulating the problem and not be worried about the enormous amount of mathematics required to obtain the desired equations of motion. In short, it reduces problems which seemingly appear hopeless to a rather routine operation.

2.4 Kinetic Energy

This section covers the kinetic energy portion of the Lagrangian. To be more specific, it discusses the use of equation B-11. An example is given on how to obtain the A matrix of equation (2.2) by use of B-11 and also the corresponding FORTRAN coding generated by the Einstein summation notation is shown.

Since nearly all of the required matrix operations for a particular problem are contained in the definition of the kinetic energy, this becomes the obvious starting point for obtaining the necessary differential expressions. It will be seen in later sections that many derivatives for other energy terms do not have to be calculated as this will already have been accomplished in the kinetic energy portion.

As mentioned previously, it is more efficient (computationally) to operate on the position vectors before dot products are actually taken. And also mentioned before was that the position vectors themselves are not really operated upon since they are in fact reduced to sums of smaller quantities (to reduce the imbedding problem) and it is these smaller quantities which are of interest. The actual break-up of the position vectors is covered in section 3.3. Because of the ease in actually combining the smaller terms to obtain the position vectors, the example given here will be concerned with only the position vectors.

Equation B-11 is rewritten for convenience.

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{q}} - \frac{\partial T}{\partial q} = \sum_{Q} M_Q \left[\vec{Q}_q \cdot \vec{Q}_{q_i q_j} \dot{q}_i \dot{q}_j + \vec{Q}_q \cdot \vec{Q}_{q_i} \ddot{q}_i\right] \tag{2.3}$$

where all quantities are defined in Appendix B. M_Q is the mass associated with the position vector Q, where Q takes on the various letters Q, R, S, etc., to distinguish each of the n position vectors and its corresponding mass.

It is seen from equation (2.3) that first and second partial derivatives with respect to the generalized coordinates are required. The subscripted variable PKE (i,j,k,l) provides a convenient bookkeeping notation for representing these derivatives, the mass from which they originate and the direction (x,y, or z) of the position vector with respect to the inertial coordinate system. From past experience this bookkeeping procedure has proven to be very adequate; it is organized, efficient and yet simple and it also conveys a maximum amount of information such that a particular quantity can be easily identified. For example, the i runs from 1 to n and refers to the particular mass and associated position vector that is being delt with; the l runs from 1 to 3 and refers to the x,y, or z direction; the j and k refer to the partial derivatives which have been taken with respect to the q_j generalized coordinate and the q_k generalized coordinate if second partials have been taken. If only one derivative has been taken, then k takes on the value 1 plus the number of generalized coordinates.

Suppose the number of generalized coordinates is eleven and the position vectors are Q, R, and S, then

$$\frac{\partial^2 \vec{Q}}{\partial q_5 \partial q_8} = \text{PKE}\,(1,5,8,L), \text{ where } L = 1,2,3$$

$$\frac{\partial^2 \vec{S}}{\partial q_5 \partial q_8} = \text{PKE}\,(3,5,8,L), \text{ where } L = 1,2,3$$

$$\frac{\partial R}{\partial q_9} = \text{PKE}\,(2,9,12,L), \text{ where } L = 1,2,3$$

An example of using equation (2.3) with the number of masses n = 2 and the number of equations k = 3 and considering only the acceleration terms of (2.3) yields the following result.

First equation:

$$M_Q \left[\vec{Q}_{q_1} \cdot \vec{Q}_{q_1} \ddot{q}_1 + \vec{Q}_{q_1} \cdot \vec{Q}_{q_2} \ddot{q}_2 + \vec{Q}_{q_1} \cdot \vec{Q}_{q_3} \ddot{q}_3\right]$$

$$+ M_R \left[\vec{R}_{q_1} \cdot \vec{R}_{q_1} \ddot{q}_1 + \vec{R}_{q_1} \cdot \vec{R}_{q_2} \ddot{q}_2 + \vec{R}_{q_1} \cdot \vec{R}_{q_3} \ddot{q}_3\right]$$

Second equation:

$$M_Q \left[\vec{Q}_{q_2} \cdot \vec{Q}_{q_1} \ddot{q}_1 + \vec{Q}_{q_2} \cdot \vec{Q}_{q_2} \ddot{q}_2 + \vec{Q}_{q_2} \cdot \vec{Q}_{q_3} \ddot{q}_3\right]$$

$$+ M_R \left[\vec{R}_{q_2} \cdot \vec{R}_{q_1} \ddot{q}_1 + \vec{R}_{q_2} \cdot \vec{R}_{q_2} \ddot{q}_2 + \vec{R}_{q_2} \cdot \vec{R}_{q_3} \ddot{q}_3\right]$$

Third equation:

$$M_Q \left[\vec{Q}_{q_3} \cdot \vec{Q}_{q_1} \ddot{q}_1 + \vec{Q}_{q_3} \cdot \vec{Q}_{q_2} \ddot{q}_2 + \vec{Q}_{q_3} \cdot \vec{Q}_{q_3} \ddot{q}_3\right]$$

$$+ M_R \left[\vec{R}_{q_3} \cdot \vec{R}_{q_1} \ddot{q}_1 + \vec{R}_{q_3} \cdot \vec{R}_{q_2} \ddot{q}_2 + \vec{R}_{q_3} \cdot \vec{R}_{q_3} \ddot{q}_3\right]$$

These three equations can be rewritten in the form

First equation:

$$\left[M_Q \vec{Q}_{q_1} \cdot \vec{Q}_{q_1} + M_R \vec{R}_{q_1} \cdot \vec{R}_{q_1}\right] \ddot{q}_1 + \left[M_Q \vec{Q}_{q_1} \cdot \vec{Q}_{q_2} + M_R \vec{R}_{q_1} \cdot \vec{R}_{q_2}\right] \ddot{q}_2 + \left[M_Q \vec{Q}_{q_1} \cdot \vec{Q}_{q_3} + M_R \vec{R}_{q_1} \cdot \vec{R}_{q_3}\right] \ddot{q}_3$$

Second equation:

$$\left[M_Q \vec{Q}_{q_2} \cdot \vec{Q}_{q_1} + M_R \vec{R}_{q_2} \cdot \vec{R}_{q_1}\right] \ddot{q}_1 + \left[M_Q \vec{Q}_{q_2} \cdot \vec{Q}_{q_2} + M_R \vec{R}_{q_2} \cdot \vec{R}_{q_2}\right] \ddot{q}_2 + \left[M_Q \vec{Q}_{q_2} \cdot \vec{Q}_{q_3} + M_R \vec{R}_{q_2} \cdot \vec{R}_{q_3}\right] \ddot{q}_3$$

Third equation:

$$\left[M_Q \vec{Q}_{q_3} \cdot \vec{Q}_{q_1} + M_R \vec{R}_{q_3} \cdot \vec{R}_{q_1}\right] \ddot{q}_1 + \left[M_Q \vec{Q}_{q_3} \cdot \vec{Q}_{q_2} + M_R \vec{R}_{q_3} \cdot \vec{R}_{q_2}\right] \ddot{q}_2 + \left[M_Q \vec{Q}_{q_3} \cdot \vec{Q}_{q_3} + M_R \vec{R}_{q_3} \cdot \vec{R}_{q_3}\right] \ddot{q}_3$$

The first row of the A matrix of equation (2.2) is

$$A(1,1) = \left[M_Q \vec{Q}_{q_1} \cdot \vec{Q}_{q_1} + M_R \vec{R}_{q_1} \cdot \vec{R}_{q_1}\right]$$

$$A(1,2) = \left[M_Q \vec{Q}_{q_1} \cdot \vec{Q}_{q_2} + M_R \vec{R}_{q_1} \cdot \vec{R}_{q_2}\right]$$

$$A(1,3) = \left[M_Q \vec{Q}_{q_1} \cdot \vec{Q}_{q_3} + M_R \vec{R}_{q_1} \cdot \vec{R}_{q_3}\right]$$

and so forth for the remaining six elements of A of this example.

The FORTRAN Coding for obtaining the elements of the A matrix for the translational part of the kinetic energy is now given. The rotational part must be added to these terms and this can be found in Appendix F which contains the FORTRAN program. Similarly the remaining coding for the kinetic energy is in appendix F.

Since A is symmetric, the upper triangular terms are calculated to reduce the number of arithmetic operations. Recalling that IEQS = 3 and IMASS = 2 yields

```
  DO 1 j = 1,IEQS
  DO 1 k = j,IEQS
  SUM = 0.
  DO 2 i = 1,IMASS
  DO 2 l = 1,3
2 SUM = SUM + XMASS(i)* PKE(i,j,12,1)* PKE(i,k,12,1)
  A(j,k) = SUM
1 A(k,j) = SUM
```

2.5 Other Energy Terms

The differential expressions required for other energy terms are either obtained from those which have already been calculated or are derived separately. As an example of how to utilize expressions already calculated, the reader can refer to equations A-7 through A-11 in Appendix A. These five equations are very similar to equation A-12 except that A-12 considers only the z direction and involves both

multiplication by the gravitational constant, g and by the five masses. Therefore, $\partial U_1/\partial q_j$ can be obtained very quickly from the following definitions (which are explained in section 3.3). For equations

A-7-11

$$\frac{\partial \vec{Q}}{\partial q_j} = \text{PKE}\ (1,j,12,3)$$

$$\frac{\partial \vec{R}}{\partial q_j} = \text{PKE}\ (2,j,12,3)$$

$$\frac{\partial \vec{S}}{\partial q_j} = \text{PKE}\ (3,j,12,3)$$

$$\frac{\partial \vec{T}}{\partial q_j} = \text{PKE}\ (4,j,12,3)$$

$$\frac{\partial \vec{U}}{\partial q_j} = \text{PKE}\ (5,j,12,3)$$

The $\partial U_1/\partial q_j$ makes up a part of the B vector in equation (2.2) which forms the right hand sides of the differential equations. The contribution of $\partial U_1/\partial q_j$ (from the potential energy) to the right hand sides (utilizing calculations from the kinetic energy) is evaluated as follows

```
      DO 14 J = 1,IEQS
   14 RHS(J) = RHS(J) + GRAV*(XMASS(1)*PKE(1,J,12,3) + XMASS(2)*
     1 PKE(2,J,12,3) + XMASS(3)*PKE(3,J,12,3) + XMASS(4)*PKE(4,J,12,3) +
     2 XMASS(5)*PKE(5,J,12,3))
```

Note that the fourth subscript of PKE is equal to 3 which signifies that only those terms in the z direction are used. The third subscript is equal to twelve since only one partial derivative is taken (the number of generalized coordinates is eleven).

As much use as possible is made of all previously defined matrix operations or whatever caculations have been performed to aid in obtaining new partial derivatives. This can readily be observed when examining the FORMAC program.

2.6 Numerical Integration

A fourth order Runge-Kutta integration scheme is used to integrate the matrix differential equation

$$A(q)\ddot{q} = B(q,\dot{q},t)$$

To solve for the $\ddot{q}$, i.e.,

$$\begin{aligned}
\ddot{q}_1 &= f_1(q_i,\dot{q}_i,t) \\
\ddot{q}_2 &= f_2(q_i,\dot{q}_i,t) \qquad (2.4) \\
&\cdot \\
&\cdot \\
&\cdot \\
\ddot{q}_k &= f_k(q_i,\dot{q}_i,t)
\end{aligned}$$

advantage is taken of the symmetry in the mass matrix A. The subroutine which decouples the acceleration terms into the form of equations (2.4) is called SOLVE and the algorithm used in the Square Root Method. Appendix D describes the modifications that were made to that method to eliminate the concern of pure imaginary numbers. A description of each subroutine and its purpose is given in the FORTRAN listing in Appendix F.

3.0 APPLICATION OF SOLUTION TECHNIQUE

3.1 Symbol Table

Computer Symbol	Corresponding Analysis Symbol	Definition	Unit
A1, A2, A3	a_1, a_2, a_3	Coordinates of the mass center of vehicle in O_2-A"B"C"	inches
AAA (11, 11)		Coefficients matrix of acceleration (mass Matrix) terms	$\frac{\text{pound second}^2}{\text{inches}}$ or lb.sec.^{2}in.
A1SUB, (A1BAR), A2SUB, A3SUB	$\underline{A_1}$, $(\overline{A}_1)$, $\underline{A_2}$, $\underline{A_3}$	Coordinates of the attachments of the brace to the vehicle in O_4 - X"Y"Z"	inches
ALPHA1, ALPHA2, ALPHA3	α_1, α_2, α_3	Coordinates of the horizontal attachment points of ground springs to spade assembly in O_1 - ABC	inches
ASTAR		Angle between ℄ of brace and a plumb line dropped from the attachment of the brace to the vehicle	degrees (changed to radians in program)
BETA	β	$\beta = \frac{\sigma}{2g} \frac{A^3}{A_0^2}$ where σ = specific weight of oil, A = area of piston and A_0 = effective area of orifice. β is a hydraulic constant for the spade cylinder	$\frac{\text{pound sec}^2}{\text{inches}^2}$
BBETAE, BETAE	β_E, β_{EMAX}	torque provided by friction brake in elevating mechanism. β_{EMAX} is maximum.	inch pounds
BBETAT, BETAT	β_τ, $\beta_{\tau MAX}$	torque provided by friction brake in traversing mechanism. $\beta_{\tau MAX}$ is maximum.	inch pounds
BOFT	B(t)	Breech force	pounds

Computer Symbol	Corresponding Analysis Symbol	Definition	Unit
BRCHX (105)		Abcissa of breech force curve(time)	seconds
BRCHY (105)		Ordinate of breech force curve	pounds
B1, (B1BAR) B2, B3	B_1 $(\bar{B}_1)$, B_2, B_3	Coordinates of the attachments of the spade cylinders to the vehicle in O_4 - X"Y"Z"	inches
CBRCE	C_i	Damping coefficient associated with K_i	$\frac{\text{pound sec}}{\text{inch}}$
CC11, CC12, CC21, CC22	C_{ij}	Damping coefficients associated with K_{ij}	$\frac{\text{pound sec}}{\text{inch}}$
COFT	C(t)	Recuperator force	pounds
CU5		Damping coefficient for use in determining initial conditions	in. lbs. sec.
C1P, C2P, C3P	C'_1, C'_2, C'_3	Coordinates of the attachments of the spade cylinders to the spade assembly in O_7 - U'V'W'	inches
DA (1, 1, 1, L) L= 1,2,3,4 1↔(1,1) 2↔(1,2) 3↔(2,1) 4↔(2,2)	δ_{ij}	Extension/contraction of vertical ground spring	inches
DP (1, 1, 1, 4)	δ'_{ij}	Spring deflections($Q_{ij}{}^{iv}$)	$\frac{\text{pounds}}{\text{inch}}$

Computer Symbol	Corresponding Analysis Symbol	Definition	Unit
D1, D2, D3	d_1, d_2, d_3	Coordinates of the center of the trunnion in $O_5 - X^{iv}Y^{iv}Z^{iv}$	inches
D1P, D2P, D3P	D_1', D_2', D_3'	Coordinates of the attachments of the braces to the spade assembly in $O_7 - U'V'W'$	inches
EB	ε_B	Distance from mass center of recoiling parts to point of application of breech force in η direction only. ε_B is negative.	inches
EC	ε_C	Distance from mass center of recoiling parts to point of application of recuperation force in η direction only. ε_C is negative.	inches
ER	ε_R	Distance from mass center of recoiling parts to point of application of recoil force in η direction only. ε_R is negative.	inches
ELANG		Angle of elevation for use in determining initial conditions of Q(7)	radians
E1, E2, E3	e_1, e_2, e_3	Coordinates of the center of the traverse bearings in $O_4 - X''Y''Z''$	inches
FOFG	$F(\gamma)$	Equilibrator force as a function of γ.	pounds

Computer Symbol	Corresponding Analysis Symbol	Definition	Unit
FF1, FF2, FF3	f_1, f_2, f_3	Coordinates of the mass center of the traversing but non-elevating parts in $O_5 - X^{iv}Y^{iv}Z^{iv}$.	inches
GAMMAX (105)		Abscissa of equilibrator force curve data in degrees but converted to radians in program.	radians
GAMMAY (105)	$F(\gamma)$	Ordinate of equilibrator force curve.	pounds
GKST	K_{st}	Effective torsional spring about trunnion used in determining initial conditions	$\frac{\text{inch pounds}}{\text{radians}}$
GRAV	g	Acceleration due to gravity	inches/sec^2
G1, G2, + Q(2), G3	g_1 $g_2 + v$, g_3	Coordinates of the center of pressure of the spades in O_3 - XYZ. Note v is a generalized coordinate.	inches
HH1, HH2, HH3	h_1, h_2, h_3	Coordinates of the mass center of the spades in O_7 - U'V'W'.	inches
HOOR1		Delta function used in defining initial conditions	
IBOFT		table lookup counter on breech force	
IBPTS		No. of ordered pairs in breech force table	
IEQS		Number of generalized coordinates	
IGOFT		Table lookup counter on equilibrator	

Computer Symbol	Corresponding Analysis Symbol	Definition	Unit
IGPTS		No. of ordered pairs of points in equilibrator table	
IMASS		Number of masses	
IROFT		Table lookup counter on recoil force.	
IRPTS		No. of ordered pairs of points in recoil force table.	
O2, O3	O_2, O_3	Coordinates of the equilibrator attachment points to the elevating but non-recoiling parts in $O_6 - \Xi'H'Z'$	inches
PD (3, 11, 12, 4)		Derivative functions in dissipative energy	
PG (6, 12, 12, 3)		Derivative function in generalized forces	
PT (8, 11, 12, 3)		Derivative function in translational part of kinetic energy	
PU (4, 11, 12, 4)		Derivative functions in potential energy	
PW (5, 12, 12, 3)		Derivative functions in rotational part of kinetic energy	
QDD (I)		Accelerations of the generalized coordinates I = 1,...,11	$\frac{\text{Units of Q(i)}}{\text{second}^2}$
QD(I)		Velocities of the generalized coordinates, I = 1,...,11	$\frac{\text{Units of Q(i)}}{\text{second}}$
QSAVE (11)		Dimension variable which saves the generalized coordinates in Runge-Kutta integration	

Computer Symbol	Corresponding Analysis Symbol	Definition	Unit
QDSAVE (11)		Dimension variable which saves generalized velocities in Runge-Kutta integration.	
Q(1)	η	Translation of recoiling parts. Has initial value.	inches
Q(2)	v	Fore (+) and aft (-) motion of spade assembly.	inches
Q(3)	x	Lateral displacement of total weapon system	inches
Q(4)	y	Fore (+) and aft(-) motion of vehicle and traversing parts	inches
Q(5)	z	Up (+) and down (-) motion of vehicle and traversing parts	inches
Q(6)	ϕ	Pitch of vehicle and traversing parts	radians
Q(7)	γ	Pitch of elevating parts relative to the vehicle. May have initial value angle of elevation = γ_o	radians
Q(8)	ν	Pitch of spade assembly	radians
Q(9)	θ	Roll of total weapon system	radians
Q(10)	ψ	Yaw of total weapon system	radians

Computer Symbol	Corresponding Analysis Symbol	Definition	Unit
Q(11)	τ	Yaw of traversing parts relative to the vehicle. May have initial value (angle of traverse = τ).	radians
RHS (11)		Right hand side of equations of motion.	
RODY (105)		Ordinate of recoil force	pounds
RODX (105)		Abcissa of recoil force (time)	seconds
ROFT	R(t)	Recoil force	pounds
TIME	t	Time measured from initiation of solution.	sec
TIMEH		Integration step size	sec
TIMEH2		Defined as TIMEH/2.	sec
TIMEH8		Defined as TIMEH/8	sec
XI, Q(1), ZETA	ξ,η,ζ	Coordinates of the mass center of the recoiling parts in $O_6 - \Xi'H'Z'$ Note η is generalized coordinate.	inches
XIB, Q(1) + EB, ZETAB	Ξ_B, η_B, ζ_B	Coordinates of point of application of breech force in $O_6 - \Xi'H'Z'$	inches

Computer Symbol	Corresponding Analysis Symbol	Definition	Unit
XIC, Q(1) + EC, ZETAC	ξ_C, η_C, ζ_C	Coordinates of point of application of recuperator force in O_6 - $\Xi'H'Z'$	inches
XIR, Q(1) + ER, ZETAR	ξ_R η_R ζ_R	Coordinates of point of application of recoil force in O_6 - $\Xi'H'Z'$	inches
XIXX(I)		Moments of inertia of masses about their principal "X" axis where Q, R, S, T, U correspond to I = 1,2,3,4,5	lbs sec^2 in
XIXY(I)		Cross products of inertia	lbs sec^2 in
XIXZ(I)		Cross products of inertia	lbs sec^2 in
XIYY(I)		Moments of inertia of masses about their principal "Y" axis where Q, R, S, T, U correspond to I = 1,2,3,4,5	lbs sec^2 in
XIYZ(I)		Cross products of inertia	lbs sec^2 in
XIZZ(I)		Moments of inertia of masses about their principal "Z" axis where Q, R, S, T, U correspond to I = 1,2,3,4,5	lbs sec^2 in
XI1, ETA1, ZETA1	ξ_1, η_1, ζ_1	Coordinates of the mass center of the elevated but non-recoiling parts in O_6 - $\Xi'H'Z'$	inches
XKK1, XKK2	K_i	Effective spring rate of one of the braces	pounds/inch

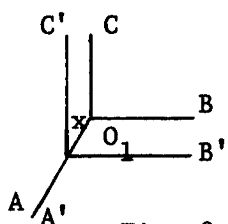
Computer Symbol	Corresponding Analysis Symbol	Definition	Unit
XKP11, XKP12, XKP21, XKP22	K_{ij}	Horizontal ground spring at four roller wheels to prevent lateral translation	pounds/inch
XKY1, XKY2	K'_i	Spring rate of horizontal ground springs associated with spade assembly.	pounds/inch
XK(1,1), XK(1,2) XK(2,1), XK(2,2)	K'_{ij}	Spring rates associated with ground springs at front and rear roller wheels	pounds/inch
XL (1,1), XM(1,1), XN(1,1), XL(1,2) XM(1,2), XN(1,2) XL(2,1), XM(2,1) XN(2,1), XL(2,2) XM(2,2), XN(2,2)	l_{ij}, m_{ij}, n_{ij}	Coordinates of ground springs at front and rear roller wheels i = 1(right) i=2(left) j = 1(front j=2(rear) in O_4 - X"Y"Z"	inches
XLENGH		Length of equilibrator	inches
XMASS(1)	M_v	Mass of the vehicle	$\frac{\text{pound second}^2}{\text{inch}}$
XMASS(2)	M_s	Mass of the spade assembly	$\frac{\text{pound second}^2}{\text{inch}}$

Computer Symbol	Corresponding Analysis Symbol	Definition	Unit
XMASS(3)	M_t	Mass of the traversing but non-elevating parts	$\frac{\text{pound second}^2}{\text{inch}}$
XMASS(4)	M_e	Mass of elevating but non-recoiling parts	$\frac{\text{pound second}^2}{\text{inch}}$
XMASS(5)	M_r	Mass of recoiling parts	$\frac{\text{pound second}^2}{\text{inch}}$
XMU	μ	Coefficient of friction between vehicle tread and ground	
XNN2,XNN3	N_2, N_3	Coordinates of the equilibrator attachment points to the traversing but non-elevating parts in $O_5 - X^{iv}Y^{iv}Z^{iv}$	inches
ZZ(I)	I = 1,2,...,90	Factored algebraic expressions to reduce the number of arithmetic operations	

3.2 Technique of Generating Energy Expressions and Generalized Forces.

The techniques used in the development of Appendix A are illustrated in the following paragraphs.

Assuming a vector to be defined in one coordinate system, it is necessary to determine a coordinate transformation which will define that vector in a different coordinate system. These coordinate transformations are needed to refer velocity and displacement vectors to a fixed coordinate system and angular velocity vectors to the appropriate body axes. As an example, three coordinate systems are defined. They are: (1) 0_1-ABC, an inertial coordinate system (fixed in space) having its origin at the mass center of the weapon system in the emplaced position; (2) 0_2-$A^1B^1C^1$, a coordinate system initially coincident with 0_1-ABC and remaining parallel to 0_1-ABC at all times where 0_2 is the center of mass of the system; and (3) 0_2-A"B"C", a coordinate system fixed in the weapon system and moving with it

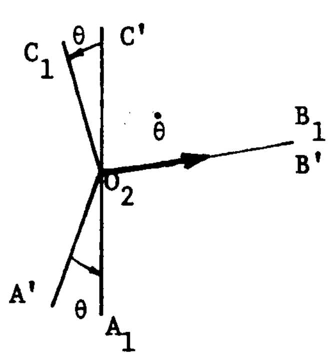


$$A = A^1 + x$$
$$B = B^1$$
$$C = C^1$$

Fig. 3.1

or

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} A^1 \\ B^1 \\ C^1 \end{bmatrix} + \lambda_1 \text{ where } \lambda_1 = \begin{bmatrix} x \\ 0 \\ 0 \end{bmatrix}$$



$$A^1 = A_1 \text{ Cos } \theta + C \text{ Sin } \theta$$
$$B^1 = B_1$$
$$C^1 = -A_1 \text{ Sin } \theta + C \text{ Cos } \theta$$

Fig. 3.2

thus

$$\begin{bmatrix} A^1 \\ B^1 \\ C^1 \end{bmatrix} = \lambda a \begin{bmatrix} A_1 \\ B_1 \\ C_1 \end{bmatrix} \quad \lambda a = \begin{bmatrix} \text{Cos } \theta & 0 & \text{Sin } \theta \\ 0 & 1 & 0 \\ -\text{Sin } \theta & 0 & \text{Cos } \theta \end{bmatrix}$$

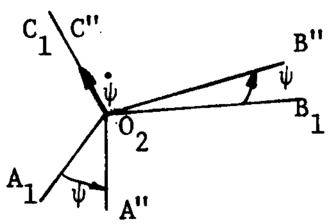


Fig. 3.3

$$A_1 = A'' \text{ Cos } \psi - B'' \text{ Sin } \psi$$

$$B_1 = A'' \text{ Sin } \psi + B'' \text{ Cos } \psi$$

$$C_1 = C''$$

and

$$\begin{bmatrix} A_1 \\ B_1 \\ C_1 \end{bmatrix} = \lambda_b \begin{bmatrix} A'' \\ B'' \\ C'' \end{bmatrix} \quad \text{where } \lambda_b = \begin{bmatrix} \text{Cos } \psi & -\text{Sin } \psi & 0 \\ \text{Sin } \psi & \text{Cos } \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and hence by successive substitution

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \lambda_1 + \lambda_2 \begin{bmatrix} A'' \\ B'' \\ C'' \end{bmatrix} \quad \text{where } \lambda_2 = \lambda_a \cdot \lambda_b$$

The matricies λ_a, λ_b, and λ_2 are orthogonal and hence their inverses are their transposes. Now if the coordinates of a point in the weapon system are known in 0_2-A"B"C" they can be determined in 0_1-ABC. All transformations in Appendix A were determined in this fashion.

The absolute angular velocity of the system about the 0_2-A"B"C" coordinate system is (see Figure 3.1 and 3.2)

$$\omega_x = \dot{\theta} \text{ sin } \psi$$

$$\omega_y = \dot{\theta} \text{ cos } \psi$$

$$\omega_z = \dot{\psi}$$

or

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}_{A''B''C''} = \lambda_b^{-1} \begin{bmatrix} 0 \\ \dot{\theta} \\ \dot{x} \end{bmatrix}$$

All angular velocities were obtained by similar transformations. Defining an element of mass in the weapon system to be dm and the vector from the origin of the fixed coordinate system to that element of mass to be $\vec{p}$, then the kinetic energy (dt) of that element of mass is

$$dT = 1/2\ \dot{\vec{p}} \cdot \dot{\vec{p}}\, dM$$

Define

$$\vec{p} = \vec{P} + \vec{\rho}$$

where $\vec{P}$ is the vector from the origin of the fixed coordinate system to the mass center of the weapon system and $\vec{\rho}$ is the vector from the mass center of the weapon system to the element of mass dm. Now

$$dT = 1/2\ (\dot{\vec{P}} + \dot{\vec{\rho}}) \cdot (\dot{\vec{P}} + \dot{\vec{\rho}})dm$$

Since the weapon system has a rigid body rotation, then

$$\dot{\vec{\rho}} = \vec{\omega} \times \vec{\rho}$$

thus

$$dT = 1/2\ (\dot{\vec{P}} + \vec{\omega} \times \vec{\rho}) \cdot (\dot{\vec{P}} + \vec{\omega} \times \vec{\rho})\, dm$$

expanding and integrating

$$T = 1/2\, M \vec{P} \cdot \vec{P} + \vec{P} \cdot \vec{\omega} \times \int_M \rho \, dm + \int_M (\vec{\omega} \times \vec{\rho}) \cdot (\vec{\omega} \times \vec{\rho}) \, dm$$

The sum of the first moments of a mass about its own mass center is zero. Thus

$$\int_M \rho \, dM = 0$$

and $\vec{\omega} \times \vec{\rho}$ can be written as;

$$(\vec{\omega} \times \vec{\rho}) = \hat{i}(\omega_y z - \omega_z y) + \hat{j}(\omega_z x - \omega_x z) + \hat{k}(\omega_x y - \omega_y x)$$

and

$$\int_M (\vec{\omega} \times \vec{\rho}) \cdot (\vec{\omega} \times \vec{\rho}) \, dM = \omega_x^2 \int_M (y^2 + z^2) \, dM + \omega_y^2 \int_M (x^2 + z^2) \, dM$$

$$+ \omega_z^2 \int_M (x^2 + y^2) \, dM - 2\omega_x \omega_y \int_M x y \, dM$$

$$- 2 \omega_x \omega_z \int_M x z \, dM - 2 \omega_y \omega_z \int_M y z \, dM$$

By definition of moment of inertia and cross products of inertia

$$1/2 \int_M (\vec{\omega} \times \vec{\rho}) \cdot (\vec{\omega} \times \vec{\rho}) \, dM = 1/2 \; (I_x \omega_x^2 + I_y \omega_y^2 + I_z \omega_z^2)$$

$$- (I_{xy} \omega_x \omega_y + I_{xz} \omega_x \omega_z + I_{yz} \omega_y \omega_z)$$

Neglecting cross products of inertia the kinetic energy of the weapon is

$$T = 1/2\, M \dot{\vec{P}} \cdot \dot{\vec{P}} + 1/2\, (I_x \omega_x^2 + I_y \omega_y^2 + I_z \omega_z^2)$$

where

$$\dot{P} = \dot{\lambda}_1$$

and the ω_i are as previously defined. I_x, I_y, I_z are the principal moments of inertia of the system about the A"B"C" axes.

The potential energy of a mass is simply the weight of that mass times the Z-coordinate of its mass center as measured in the fixed coordinate system. Define

$\lambda_0 = [0\ 0\ 1]$, then for our example

$$U_1 = \lambda_0 \cdot \lambda_1$$

Energy stored in a spring having a spring constant K is

$$U_2 = 1/2\ K\ (\Delta L)^2$$

where ΔL is the extension/contraction of the spring. It is only necessary to define L as a function of the generalized coordinates.

Associated with springs in a system is damping. Thus for the Spring K assume a damping coefficient C. Then the dissipative function for damping corresponding to the energy function for the spring is

$$\overline{U}_2 = 1/2\, C\ (\Delta \dot{L})^2$$

Assuming a forcing function, F(t) to be applied at a_1, a_2, a_3 in 0_2 -A"B"C" then in 0_1 - ABC the point of application is

$$\begin{bmatrix} A_f \\ B_f \\ C_f \end{bmatrix} = \lambda_1 + \lambda_2 \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

The direction of the force in 0_1 - ABC is

$$\lambda_2 \begin{bmatrix} F_1(t) \\ F_2(t) \\ F_3(t) \end{bmatrix}$$

The generalized force is then written as

$$Q_i = \lambda_2 \begin{bmatrix} F_1(t) \\ F_2(t) \\ F_3(t) \end{bmatrix} \cdot \frac{\partial}{\partial qi} \left\{ \lambda_1 + \lambda_2 \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \right\}$$

where i refers to the particular generalized coordinate q.

There are four basic moments acting on the elevating parts around the trunnion: (1) the moment due to the weight of the elevating parts, (2) the moment due to the equilibrator, (3) the moment due to the breech force, and (4) a moment due to the friction brake. The friction brake will offer sufficient torque to cancel out the other three moments and no pitch motion occurs until the algebraic sum of the other three moments exceeds the maximum torque that can be generated by the friction brake. Then the brake will offer a constant (maximum) resistance until the pitch velocity becomes zero. This is the argument used to generate the logic for the mathematical simulation (in Appendix A) of the friction brake.

3.3 FORMAC Procedure

This section describes the procedure that was used to obtain the necessary partial derivatives to solve Lagrange's equation of motion for those energy expressions derived in Appendix A by utilizing the results of Appendix B. The translational part of the kinetic energy will be given first, followed by the remaining energy expressions.

Equations (A-7) through (A-11) of Appendix A can be obtained from the definitions given below by summing combinations of the appropriate expressions.

$$PT(1,L) = \lambda_1 + \lambda_2\lambda_3$$

$$PT(2,L) = \lambda_2\lambda_4$$

$$PT(3,L) = \lambda_2\lambda_{10} + \lambda_2\lambda_{11} \begin{bmatrix} h_1 \\ h_1 \\ h_1 \end{bmatrix}$$

$$PT(4,L) = \lambda_2\lambda_5\lambda_6$$

$$PT(5,L) = \lambda_2\lambda_5\lambda_7 \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

$$PT(6,L) = \lambda_2\lambda_5\lambda_7\lambda_8$$

$$PT(7,1) = \lambda_2\lambda_5\lambda_7\lambda_9 \begin{bmatrix} \xi_1 \\ \eta_1 \\ \zeta_1 \end{bmatrix}$$

$$PT(8,L) = \lambda_2\lambda_5\lambda_7\lambda_9 \begin{bmatrix} \xi \\ \eta \\ \zeta \end{bmatrix}$$

where L = 1,2,3

It is more efficient to differentiate these eight quantities rather than differentiate the five original position vectors $\vec{Q},\vec{R},\vec{S},\vec{T}$ and $\vec{U}$ as repeated terms will expand the derivatives into very large expressions. By working with the smaller quantities, the analyst is able to quickly reduce the size of the partial derivatives utilizing the replace statement in the FORMAC program, see Appendix E. Also, computational costs are higher if differentiation is performed on the original expressions as considerably more core storage is necessary and, in general, the expressions just become too large to handle efficiently. The entire FORMAC program for this problem required 230K bytes of core using TSO. However, in order to keep the core size at a minimum, quantities that weren't needed for certain computations were commented out.

The break-up of the five position vectors into the eight smaller vectors appear exactly the same here as in the FORMAC program, see Appendix C. After the definition of these quantities has been made, the differentiation can proceed. As required by equation B-11, first and second partials are to be taken. Once this has been accomplished the expressions are punched in FORTRAN format. The first and second partials of these eight expressions are defined by the subscripted variable PT(1,J,K,L), ... , PT(8,J,K,L) where J = 1,2, ... , 11 and represents the partial derivative with respect to the J^{th} generalized coordinate; K = 1,2, ... , 12 and represents the partial derivative with respect to the K^{th} generalized coordinate except when K = 12. This signifies only one derivative has been taken; L = 1,2,3 for the x,y, or z direction.

Near the beginning of subroutine NAME in the FORTRAN program, the differential expressions PT(I,J,K,L) are combined to give the partial derivatives of the original position vectors. These derivatives have the variable name PKE(I,J,K,L) where I = 1,2,3,4,5, for the five position vectors and the J,K, and L run to 11, 12, and 3 respectively.

The next energy term to be discussed is the kinetic energy (angular part). Equation (B-14) determines the derivatives that are needed for the rotational part of the kinetic energy. Since the inertia terms are constant in the A and B vectors of equation (B-14), all that is necessary at the moment is to define the quantities $W_x(i)$, $W_y(i)$, and $W_z(i)$ where i = 1,2,3,4,5 for the five masses and then differentiate these terms. This procedure begins on line 2390 in the FORMAC program of Appendix C. Because equations (A-5) and (A-6) are equal, only four angular vector quantities need to be differentiated. The definition of the four quantities begins on line 2650, however, the development starts on line 2390 and because of the ease in following the program, it is not necessary to discuss the matrix operations defining the angular velocity expressions.

The inertia terms are combined with the partial derivatives of the angular terms in the FORTRAN program. Also, the coefficients of the acceleration terms are added to the acceleration coefficients of the translational part. The remaining terms are added into the right hand sides. Because equations (A-5) and (A-6) were equal, it was not necessary to calculate the partials in (A-6). However, their corresponding inertia terms have different numerical values and must be combined appropriately. Therefore, equation (A-6) is set equal to equation (A-5) in subroutine DER2 so that this can be accomplished. Thus the subscripted variable for the masses now runs from 1 to 5.

Since differentiation with respect to the generalized velocities is required, the definition of the subscripted variable changes for the angular quantities. These are as follows for the variable PW(I,J,K,L). I = 1,2,3,4,5 and defines the I^{th} mass that is being delt with. J = 1,2,...,12 and defines which partial derivative has been taken with respect to the J^{th} generalized coordinate. K = 1,2,...,12 and defines which partial derivative has been taken with respect to the K^{th} generalized velocity. L = 1,2,3 for the x,y or z

direction. When J = 12 and/or K = 12, this indicates no partial has been taken with respect to the J^{th} generalized coordinate or the K^{th} generalized velocity or both. An examination of equation (B-14) shows both the A and B vectors are required with no derivatives. The derivatives for equation (B-15) already exist from those taken in equation (B-14). Since there are not any similar terms in these two equations, the negative of (B-15) is required as was not the case in the translational analysis.

The partial derivatives of the subscripted variable OMEGA (I,L), which begins on line 2650, are defined to be PW(I,J,K,L).

A discussion of the derivatives for the potential energy will now take place. The potential energy function, PE, is defined to be $PE = \sum_{i=1}^{6} U_i$, see equation (A-15). As required by equation (2.1), the partial derivative of each of the six components of PE with respect to the generalized coordinates must be taken. The $\partial U_1/\partial q_j$ did not actually habe to be performed as all of the required derivatives have already been calculated from the kinetic energy. This is discussed in section 2.5

The derivatives of U_2, U_3, and U_4 can be followed in the FORMAC program starting at statement number 3710. In statement number 3860, LAM25 (1,KKK) was LAM25 (3,KKK) when $\partial U_2/\partial q_j$ was taken. The 3 was changed to a 1 for the partial differentiation of $\bar{U}_1$ in the dissipative energy. Working with TSO, this change was very simple as compared to coding additional statements. This short cut is rather unfortunate as the step by step procedure in the FORMAC program is not entirely in sequence. Only 230K bytes of core was allowed and thus every possible use was made of every previous operation.

The subscripted variable PU(I,J,K,L) for the potential energy has the following definitions for the subscripts. I = 1,2,3,4 and refers to U_1 through U_4. U_5 and U_6 are added to the right hand sides in subroutine NAME. J = 1,2,...,11 and represents differentiation with respect to the J^{th} generalized coordinate. K = 12 since second

partials are not taken. L = 1,2,3,4 for the U_2 as this term was broken up into 4 terms due to the 4 ground springs, see definition of variable DA(I,J,K,L) in section 3.1. U_3 and U_4 were each performed in two parts as indicated by equations (A-13) and (A-14).

The dissipative energy function, DE, is defined as DE = $\bar{U}_1 + \bar{U}_2 + \bar{U}_3$ where $\bar{U}_1, \bar{U}_2$ and $\bar{U}_3$ are given in equations (A-16,17,18). The DE function requires differentiation with respect to the generalized velocities. Thus, for $\bar{U}_1$ in equation (A-13), let EXP stand for the expression involving the sums and products of the matrices. Then,

$$\bar{U}_1 = \frac{1}{2} C_{ij} \left[\frac{d}{dt} \text{ EXP} \right]^2$$

Consider now

$$\frac{d}{dt} \text{ EXP} = \frac{\partial \text{EXP}}{\partial q_1} \dot{q}_1 + \frac{\partial \text{EXP}}{\partial q_2} \dot{q}_2 + \cdots + \frac{\partial \text{EXP}}{\partial q_n} \dot{q}_n$$

$$\frac{\partial \bar{U}_1}{\partial \dot{q}_j} = \frac{\partial}{\partial \dot{q}_j} \left[\frac{1}{2} C_{ij} \left[\frac{\partial \text{EXP}}{\partial q_1} \dot{q}_1 + \cdots + \frac{\partial \text{EXP}}{\partial q_n} \dot{q}_n \right]^2 \right]$$

$$= C_{ij} \left[\frac{\partial \text{EXP}}{\partial q_1} \dot{q}_1 + \cdots + \frac{\partial \text{EXP}}{\partial q_n} \dot{q}_n \right] \frac{\partial \text{EXP}}{\partial q_j}$$

Therefore, only the $\partial \text{EXP}/\partial q_j$ is calculated and the sums and products of terms are performed in subroutine NAME. The utilization of the calculations performed in the potential energy were taken advantage of in obtaining the expression EXP.

The necessary derivatives for the dissipative function $\bar{U}_2 = \frac{1}{2} C_i (\dot{L}_2)^2$ may be obtained by the following analysis.

$$\frac{d}{dt} L_2 = \frac{\partial L_2}{\partial q_1} \dot{q}_1 + \frac{\partial L_2}{\partial q_2} \dot{q}_2 + \cdots + \frac{\partial L_2}{\partial q_n} \dot{q}_n$$

$$\frac{\partial \bar{U}_2}{\partial \dot{q}_k} = C_i\ (\dot{L}_2)\frac{\partial \dot{L}_2}{\partial \dot{q}_k} = C_i\ (\dot{L}_2)\ \frac{\partial L_2}{\partial q_k}$$

Thus, only $\partial L_2/\partial q_k$ needs to be determined. $C_i(\dot{L}_2)\frac{\partial L_2}{\partial q_k}$ is assembled in subroutine NAME.

The same analysis is performed for $\bar{U}_3$. That is,

$$\frac{\partial \bar{U}_3}{\partial \dot{q}_k} = \beta(\dot{L}_3)^2 \quad \frac{\partial L_3}{\partial q_k}$$

where

$$\dot{L}_3 = \frac{\partial L_3}{\partial q_1}\dot{q}_1 + \frac{\partial L_3}{\partial q_2}\dot{q}_2 + \cdots + \frac{\partial L_3}{\partial q_n}\dot{q}_n$$

The required derivatives for the generalized forces are defined in section A6 of Appendix A. The reader can easily follow through the FORMAC program by refering to section A6 and subroutine NAME.

4.0 CONCLUSIONS AND RECOMMENDATIONS

This report exhibits the techniques of utilizing FORMAC for the generation of the equations of motion of a complex weapon system. The development of this technique was a major objective of this effort. It allows the analyst to spend considerably more time formulating the problem and not be too concerned about the enormous amount of mathematics to be performed. It also allows a completely nonlinear model to be developed and thus it is no longer necessary to linearize the generalized coordinates to achieve simplification of expressions. The procedure of linearizing is very time consuming, nearly impossible for large degree of freedom systems, and the risk of generating numerous errors is quite high. Utilizing FORMAC, the differential expressions are punched on cards in FORTRAN format which eliminates any potential key punching errors.

A computer model has been developed for the simulation and is operational. The model output appears reasonable based on current data available and a knowledge of how the system performs under dynamic conditions of firing. Due to the impending closure of Rodman Laboratory it was necessary that this report be written in a limited time frame since both authors were leaving the laboratory. The work presented in this report was initiated during the month of October, 1976 and was terminated in January, 1977. Therefore, some refinements and corrections to the model which would have been made if time permitted are discussed below.

A better understanding of the friction brakes in both the elevating and traversing mechanisms and an improved logic criteria is desired. The value β_{EMAX} may be a "break away" torque and under dynamic conditions is inaccurate. A more desirable logic might be

$$\text{if} \qquad \left|\frac{d}{dt}\,(I\dot{\gamma})\right| < \beta_{EMAX} \text{ then } \ddot{\gamma} = \dot{\gamma} = 0 \text{ so } \Delta\gamma = 0$$

if $\left| \frac{d}{dt} (I\dot{\gamma}) \right| > \beta_{EMAX}$ then $\beta_E = \beta_{EMAX}$ (Signium $\dot{\gamma}$)

where $I = I_{TX} + I_{UX} + M_R (\eta^2 + \zeta^2) + M_E(\eta_1^2 + \zeta_1^2)$

A more accurate value of β_E may be obtained when test data becomes available.

At the time of this analysis, only the total resisting force, R(t) + C(t), was available and was defined to be R(t). These forces, R(t) and C(t), should be individually determined as functions of the generalized coordinates to permit parametric variation studies on orifice area effects and to justify optimization design studies as meaningful. The computer program has been written to accommodate these changes.

Determination of the hydraulic force coefficient (β) resulted in such a large force (due to the exceedingly small orifice area) that it exceeded the force expected from deformation of the cylinder. Therefore, the spade cylinder was ignored in the computer runs by setting $\beta = 0$. However, if modification to the spade cylinder are made to allow the cylinder to act as a shock absorber, the design of the cylinder may be evaluated by this model.

Initial conditions may be determined in two different ways: (1) set $\frac{\partial P.E.}{\partial q_i} = 0$ and solve the resulting system of algebraic equations for initial values of the generalized coordinates or (2) assign a dashpot with each spring and without applying B(t), let the computer solve the system of equations until equilibrium is obtained. The resulting values of the generalized coordinates are then used as initial conditions. Due to time limitations, a simple static analysis was used to define initial conditions for the sample output of Appendix F.

A secondary output from the program is the dynamic loading on the various components. Spring loads are directly determined from the program. However, to determine interaction loads on various

components, free body diagrams will have to be developed and equations of motion written for the components. Since accelerations of the components are known from the program, it only remains to solve for the forces which should then be compared with the results from experimental test data. Finally, the model should be validated against data obtained from field tests which are to take place during FY 77.

References

1. Faddeeva, V.N., "Computational Methods of Linear Algebra," Dover Publications, New York, 1959

2. Unpublished M110 Vehicle Study, Pacific Car and Foundry Company, Jan 1976

3. IBM Corporation, PL/I - FORMAC Symbolic Mathematics Interpreter, Hawthorne, New York, 1969

DISTRIBUTION

Commander
U.S. Army Armament Materiel Readiness Command
Rock Island, IL 61201
1 ATTN: DRSAR-LEM
1 ATTN: DRSAR-ASA
1 ATTN: DRSAR-SA
1 ATTN: DRCPM-M110E2-TM
1 ATTN: DRCPM-CAWS

Commander
U.S. Army Test Evaluation Command
Aberdeen Proving Ground, MD 21005
1 ATTN: AMSTE-FA

Commander
Aberdeen Proving Grounds
Aberdeen Proving Ground, MD 21005
1 ATTN: STEAP-MT-A

Commander
U.S. Army Materiel Command
5001 Eisenhower Avenue
Alexandria, VA 22304
1 ATTN: AMCRD-WC

Commander
Watervliet Arsenal
Watervliet, NY 12189
1 ATTN: SWEWV-RDD
1 ATTN: Tech. Library

1 President
U.S. Army Artillery Board
Fort Sill, OK 73503

Commander
U.S. Army Armament Research and Development Command
Dover, NJ 07801
2 ATTN: DRDAR-LCW-S
1 ATTN: DRDAR-LCW-T
1 ATTN: DRDAR-LCW-A

Commander
U.S. Army Field Artillery School
Fort Sill, OK 73503
1 ATTN: ATSF-MM
1 ATTN: ATSF-CRD-WC

12 Commander
Defense Documentation Center
Cameron Station, Bldg. 5
Alexandria, VA 22134

Commander
U.S. ARmy Field Artillery School
Fort Sill, OK 73503
1 ATTN: ATSF-CD-C, J. Porter

APPENDIX A

Derivation of Energy Expressions and Generalized Forces for M110 Self-Propelled 8" Howitzer

APPENDIX A

Derivation of Energy Expressions and Generalized Forces for M110 Self-Propelled 8" Howitzer

A1. Coordinate Systems

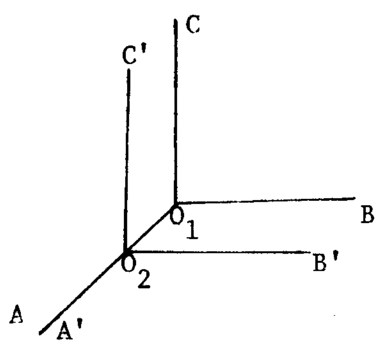
O_1 - ABC	an inertial coordinate system having its origin at the mass center of the weapons system combination in the firing position. The O_1 - AB plane is horizontal and parallel to the earth's surface. The O_1 - BC plane is vertical.
O_2 - A'B'C'	a coordinate system having its origin fixed at the center of mass of the weapons system combination. This coordinate system is at all times parallel to O_1 - ABC. The coordinates of O_2 in O_1 - ABC are (x, 0, 0).
O_2 - A"B"C"	a coordinate system reflecting the roll (θ) and yaw (ψ) of the weapons system combination. The coordinates of the mass center of the vehicle (O_3) are located in this coordinate system by (a_1, a_2, a_3).
O_3 - XYZ	a coordinate system having its origin at the mass center of the vehicle prior to the y and z translations of the vehicle and is at all times parallel to O_2 - A"B"C".
O_4 - X'Y'Z'	a coordinate system having its origin fixed at the mass center of the vehicle. This coordinate system is at all times parallel to O_3 - XYZ. The coordinates of O_4 in O_3 - XYZ are (0, y, z).

O_4 - X"Y"Z"	a coordinate system fixed in the vehicle and reflecting the pitch (ϕ) of the vehicle. The end points of the braces and spade cylinders are located in this coordinate system by $(\underline{A_1}, \underline{A_2}, \underline{A_3})$, $(\overline{A_1}, \underline{A_2}, \underline{A_3})$ and (B_1, B_2, B_3), $(\overline{B_1}, B_2, \overline{B_3})$ respectively.
O_5 - X'''Y'''Z'''	a coordinate system having its origin at the center of the traverse bearings; O_5 - Z''' being the center of traverse. This coordinate system is at all times parallel to O_4 - X"Y"Z". The coordinates of O_5 are e_1, e_2, e_3 in the O_4 - X"Y"Z" coordinate system.
O_5 - $X^{iv}Y^{iv}Z^{iv}$	a coordinate system that reflects the traverse angle, τ. The coordinates of the mass center of the traversing but non-elevating parts are f_1, f_2, f_3 in this coordinate system.
O_6 - ΞHZ	a coordinate system having its origin fixed at the midpoint between the trunnions. This coordinate system is at all times parallel to O_5 - $X^{iv}Y^{iv}Z^{iv}$.
O_6 - Ξ'H'Z'	a coordinate system that reflects the pitch of the gun (γ) relative to the understructure. Note that the initial value of γ is the angle of elevation. The coordinates of the mass center of the recoiling parts are ξ,η,ζ in this coordinate system. The coordinates of the mass center of the elevated but non-recoiling parts are ξ_1,η_1,ζ_1.

O_7 - UVW — a coordinate system having its origin at the center of pressure of the spades. The coordinates of O_7 in O_3 - XYZ are (g_1, g_2+v, g_3).

O_7 - U'V'W' — a coordinate system fixed in the spade and reflecting the pitch (ν) of the spade. The coordinates of the mass center of the spade are (h_1, h_2, h_3) in this coordinate system. The coordinates of the end point of the spade cylinders and braces attached to the spade are $(\pm C_1, C_2, C_3)$ and $(\pm D_1', D_2', D_3')$ respectively.

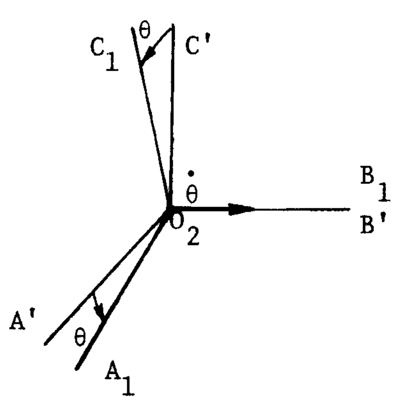
A2. Coordinate Transformations



$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} A' \\ B' \\ C' \end{bmatrix} + \lambda_1$$

where

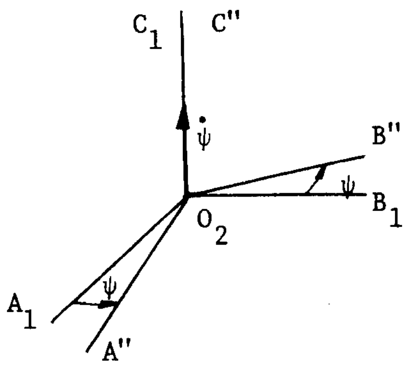
$$\lambda_1 = \begin{bmatrix} x \\ 0 \\ 0 \end{bmatrix}$$



$$\begin{bmatrix} A' \\ B' \\ C' \end{bmatrix} = \lambda_a \begin{bmatrix} A_1 \\ B_1 \\ C_1 \end{bmatrix}$$

where

$$\lambda_a = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$



$$\begin{bmatrix} A_1 \\ B_1 \\ C_1 \end{bmatrix} = \lambda_b \begin{bmatrix} A'' \\ B'' \\ C'' \end{bmatrix}$$

where

$$\lambda_b = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

So

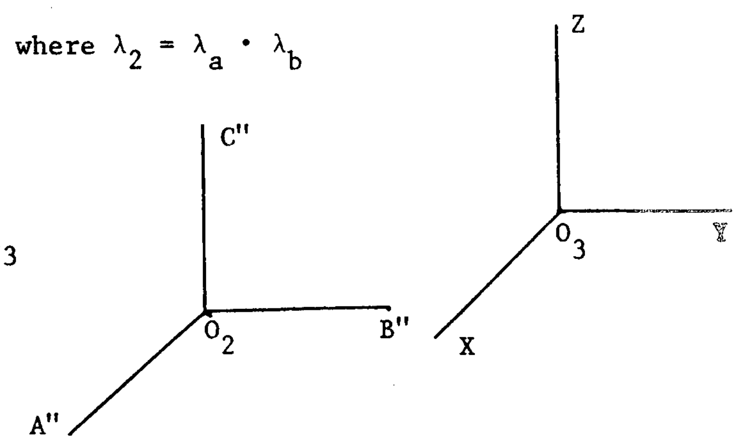
$$\begin{bmatrix} A' \\ B' \\ C' \end{bmatrix} = \lambda_2 \begin{bmatrix} A'' \\ B'' \\ C'' \end{bmatrix} \quad \text{where } \lambda_2 = \lambda_a \cdot \lambda_b$$

and

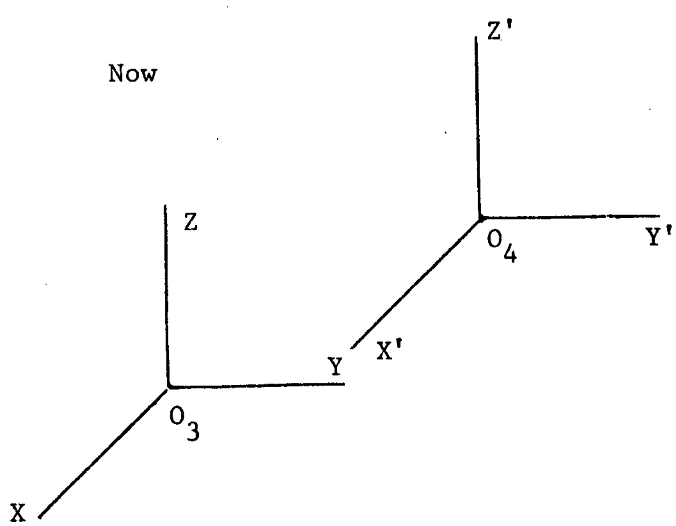
$$\begin{bmatrix} A'' \\ B'' \\ C'' \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \lambda_3$$

where

$$\lambda_3 = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$



Now

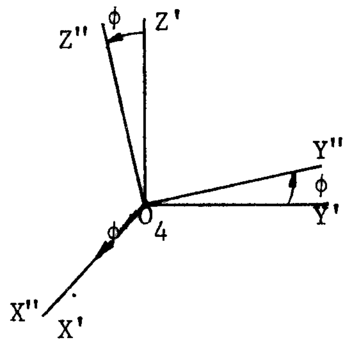


$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} + \lambda_4$$

where

$$\lambda_4 = \begin{bmatrix} 0 \\ y \\ z \end{bmatrix}$$

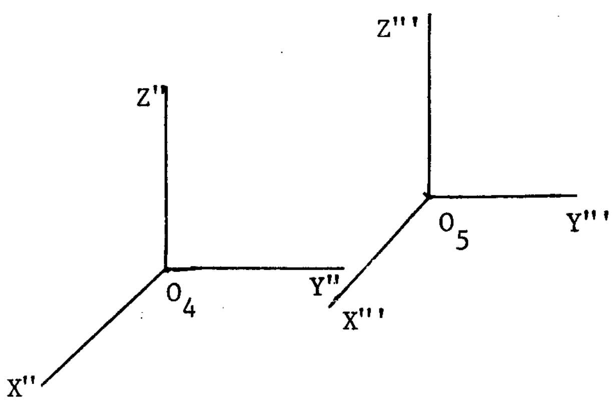
and



$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \lambda_5 \begin{bmatrix} X'' \\ Y'' \\ Z'' \end{bmatrix}$$

where

$$\lambda_5 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

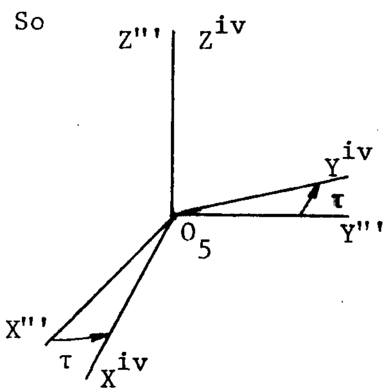


$$\begin{bmatrix} X'' \\ Y'' \\ Z'' \end{bmatrix} = \begin{bmatrix} X''' \\ Y''' \\ Z''' \end{bmatrix} + \lambda_6$$

where

$$\lambda_6 = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix}$$

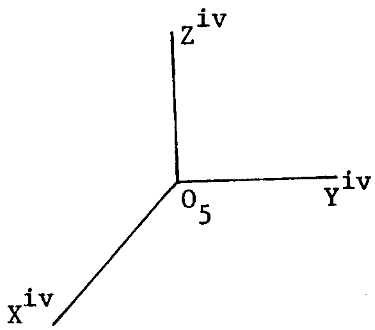
So



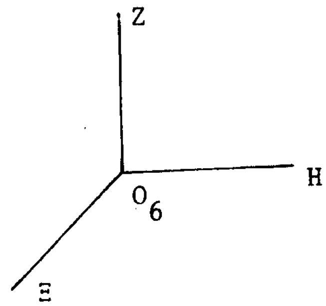
$$\begin{bmatrix} X''' \\ Y''' \\ Z''' \end{bmatrix} = \lambda_7 \begin{bmatrix} X^{iv} \\ Y^{iv} \\ Z^{iv} \end{bmatrix}$$

where

$$\lambda_7 = \begin{bmatrix} \cos\tau & -\sin\tau & 0 \\ \sin\tau & \cos\tau & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



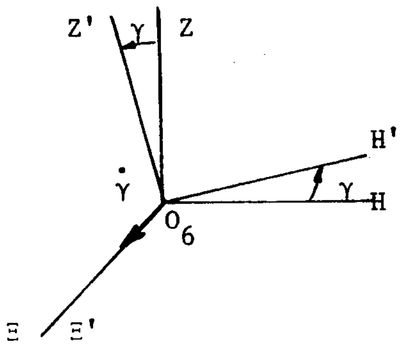
$$\begin{bmatrix} X^{iv} \\ Y^{iv} \\ Z^{iv} \end{bmatrix} = \begin{bmatrix} \Xi \\ H \\ Z \end{bmatrix} + \lambda_8$$



where

$$\lambda_8 = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}$$

and

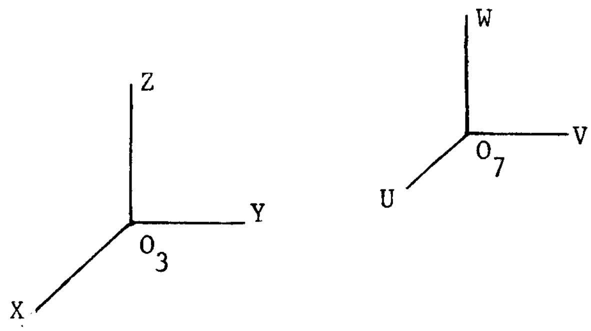


$$\begin{bmatrix} \Xi \\ H \\ Z \end{bmatrix} = \lambda_9 \begin{bmatrix} \Xi' \\ H' \\ Z' \end{bmatrix}$$

where

$$\lambda_9 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix}$$

Now

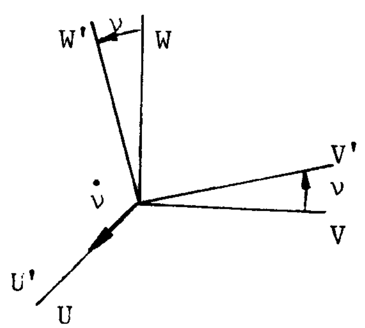


$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} U \\ V \\ W \end{bmatrix} + \lambda_{10}$$

where

$$\lambda_{10} = \begin{bmatrix} g_1 \\ g_2 + v \\ g_3 \end{bmatrix}$$

and



$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \lambda_{11} \begin{bmatrix} U' \\ V' \\ W' \end{bmatrix}$$

where

$$\lambda_{11} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \nu & -\sin \nu \\ 0 & \sin \nu & \cos \nu \end{bmatrix}.$$

A3. Kinetic Energy

Define

M_{vs}	:	Mass of vehicle/spade combination
M_v	:	Mass of vehicle
M_s	:	Mass of spade
M_t	:	Mass of traversing but non-elevating parts
M_e	:	Mass of elevating but non-recoiling parts
M_r	:	Mass of recoiling parts

Define the coordinates of the mass center of M_i

M_{vs}	:	$x, 0, 0$	in O_1 - ABC
M_v	:	a_1, a_2, a_3	in O_2 - A"B"C"
M_s	:	h_1, h_2, h_3	in O_7 - U'V'W'
M_t	:	$f_1, f_2. f_3$	in O_5 - $X^{iv}Y^{iv}Z^{iv}$
M_e	:	ξ_1, η_1, ζ_1	in O_6 - Ξ'H'Z'
M_r	:	$\xi, \eta, \zeta,$	in O_6 - Ξ'H'Z'

Define the vectors in the O_1 - ABC coordinate system from O_1 to the mass center of M_i as

M_{vs}	:	$\vec{P}$
M_v	:	$\vec{Q}$
M_s	:	$\vec{R}$
M_t	:	$\vec{S}$
M_e	:	$\vec{T}$
M_r	:	$\vec{U}$

and the vectors to an element in M_i as

$M_{vs} \quad : \quad \vec{p}$

$M_v \quad : \quad \vec{q}$

$M_s \quad : \quad \vec{r}$

$M_t \quad : \quad \vec{s}$

$M_e \quad : \quad \vec{t}$

$M_r \quad : \quad \vec{u}$

Define $\vec{\rho}$ as the vector from the mass center of M_i to the element of M_i, e.g.

$$\vec{Q} + \vec{\rho}_Q = \vec{q}$$

The angular velocities are determined as follows

$$\begin{bmatrix} \omega_{px} \\ \omega_{py} \\ \omega_{pz} \end{bmatrix}_{A''B''C''} = \lambda_b^{-1} \begin{bmatrix} 0 \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{A-1}$$

and

$$\begin{bmatrix} \omega_{qx} \\ \omega_{qy} \\ \omega_{qz} \end{bmatrix}_{X''Y''Z''} = \lambda_5^{-1} \begin{bmatrix} \omega_{px} \\ \omega_{py} \\ \omega_{pz} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} \tag{A-2}$$

Also

$$\begin{bmatrix} \omega_{rx} \\ \omega_{ry} \\ \omega_{rz} \end{bmatrix}_{U'V'W'} = \lambda_{11}^{-1} \begin{bmatrix} \omega_{px} \\ \omega_{py} \\ \omega_{pz} \end{bmatrix} + \begin{bmatrix} \dot{\nu} \\ 0 \\ 0 \end{bmatrix} \tag{A-3}$$

Now

$$\begin{bmatrix} \omega_{sx} \\ \omega_{sx} \\ \omega_{sx} \end{bmatrix}_{X^{iv}Y^{iv}Z^{iv}} = \lambda_t^{-1} \begin{bmatrix} \omega_{qx} \\ \omega_{qy} \\ \omega_{qz} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\tau} \end{bmatrix} \tag{A-4}$$

and

$$\begin{bmatrix} \omega_{tx} \\ \omega_{ty} \\ \omega_{tz} \end{bmatrix}_{\Xi'H'Z'} = \lambda_9^{-1} \begin{bmatrix} \omega_{sx} \\ \omega_{sy} \\ \omega_{sz} \end{bmatrix} + \begin{bmatrix} \dot{\gamma} \\ 0 \\ 0 \end{bmatrix} \tag{A-5}$$

Finally

$$\begin{bmatrix} \omega_{ux} \\ \omega_{uy} \\ \omega_{uz} \end{bmatrix}_{\Xi'H'Z'} = \begin{bmatrix} \omega_{tx} \\ \omega_{ty} \\ \omega_{tz} \end{bmatrix} \tag{A-6}$$

The kinetic energy differential is written as

$$dT = \tfrac{1}{2}\, \dot{\vec{q}} \cdot \dot{\vec{q}}\, dM_v + \tfrac{1}{2}\, \dot{\vec{r}} \cdot \dot{\vec{r}}\, dM_s + \tfrac{1}{2}\, \dot{\vec{s}} \cdot \dot{\vec{s}}\, dM_t + \tfrac{1}{2}\, \dot{\vec{t}} \cdot \dot{\vec{t}}\, dM_e + \tfrac{1}{2}\, \dot{\vec{u}} \cdot \dot{\vec{u}}\, dM_r$$

Since

$$\vec{q} = \vec{Q} + \vec{\rho}_Q$$

then

$$\dot{\vec{q}} = \dot{\vec{Q}} + \dot{\vec{\rho}}_Q$$

but

$$\dot{\vec{\rho}}_Q = \vec{\omega}_Q \times \vec{\rho}_Q$$

so

$$\dot{\vec{q}} = \dot{\vec{Q}} + \vec{\omega}_Q \times \vec{\rho}_Q$$

Now

$$\int_{M_v} \tfrac{1}{2}\, \dot{\vec{q}} \cdot \dot{\vec{q}}\, dM_v = \tfrac{1}{2} \int_{M_v} \dot{\vec{Q}} \cdot \dot{\vec{Q}} + 2\dot{\vec{Q}} \cdot \vec{\omega}_Q \times \vec{\rho}_Q + (\vec{\omega}_Q \times \vec{\rho}_Q) \cdot (\vec{\omega}_Q \times \vec{\rho}_Q)\quad dM_v$$

or

$$\int_{M_v} \tfrac{1}{2}\, \dot{\vec{q}} \cdot \dot{\vec{q}}\, dM_v = \tfrac{1}{2}\, \dot{\vec{Q}}\, \dot{\vec{Q}}\, M_v + \dot{\vec{Q}} \cdot \vec{\omega}_Q \int_{M_v} \vec{\rho}_Q\, dM_v + \tfrac{1}{2} \int_{M_v} (\vec{\omega}_Q \times \vec{\rho}_Q) \cdot (\vec{\omega}_Q \times \vec{\rho}_Q)\, dM_v$$

but

$$\int_{M_v} \vec{\rho}_Q\, dM_v = 0$$

and neglecting cross products of inertia

$$\tfrac{1}{2} \int_{M_v} (\vec{\omega}_Q \times \vec{\rho}_Q) \cdot (\vec{\omega}_Q \times \vec{\rho}_Q)\, dM_v = \tfrac{1}{2}\, (I_{QX}\, \omega_{QX}^2 + I_{QY}\, \omega_{QY}^2 + I_{QZ}\, \omega_{QZ}^2)$$

So

$$\begin{aligned} KE = {} & \tfrac{1}{2} M_v \dot{Q}^2 + \tfrac{1}{2} M_s \dot{R}^2 + \tfrac{1}{2} M_t \dot{S}^2 + \tfrac{1}{2} M_e \dot{T}^2 + \tfrac{1}{2} M_r \dot{U}^2 \\ & + \tfrac{1}{2} (I_{QX}\, \omega_{QX}^2 + I_{QY}\, \omega_{QY}^2 + I_{QZ}\, \omega_{QZ}^2) + \tfrac{1}{2} (I_{RX}\, \omega_{RX}^2 + I_{RY}\, \omega_{RY}^2 + I_{RZ}\, \omega_{RZ}^2) \\ & + \tfrac{1}{2} (I_{SX}\, \omega_{SX}^2 + I_{SY}\, \omega_{SY}^2 + I_{SZ}\, \omega_{SZ}^2) + \tfrac{1}{2} (I_{TX}\, \omega_{TX}^2 + I_{TY}\, \omega_{TY}^2 + I_{TZ}\, \omega_{TZ}^2) \\ & + \tfrac{1}{2} (I_{UX}\, \omega_{UX}^2 + I_{UY}\, \omega_{UY}^2 + I_{UZ}\, \omega_{UZ}^2) \end{aligned}$$

where ω_{ij} is the absolute angular velocity of the i^{th} body around the j axis of that coordinate system fixed in the body.

Now

$$\begin{bmatrix} Q_a \\ Q_b \\ Q_c \end{bmatrix} = \lambda_1 + \lambda_2 (\lambda_3 + \lambda_4) \tag{A-7}$$

$$\begin{bmatrix} R_a \\ R_b \\ R_c \end{bmatrix} = \lambda_1 + \lambda_2 \left(\lambda_3 + \lambda_{10} + \lambda_{11} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} \right) \tag{A-8}$$

$$\begin{bmatrix} S_a \\ S_b \\ S_c \end{bmatrix} = \lambda_1 + \lambda_2 \left(\lambda_3 + \lambda_4 + \lambda_5 \left(\lambda_6 + \lambda_7 \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \right) \right) \tag{A-9}$$

$$\begin{bmatrix} T_a \\ T_b \\ T_c \end{bmatrix} = \lambda_1 + \lambda_2 \left(\lambda_3 + \lambda_4 + \lambda_5 \left(\lambda_6 + \lambda_7 \left(\lambda_8 + \lambda_9 \begin{bmatrix} \xi_1 \\ \eta_1 \\ \zeta_1 \end{bmatrix} \right) \right) \right) \tag{A-10}$$

$$\begin{bmatrix} U_a \\ U_b \\ U_c \end{bmatrix} = \lambda_1 + \lambda_2 \left(\lambda_3 + \lambda_4 + \lambda_5 \left(\lambda_6 + \lambda_7 \left(\lambda_8 + \lambda_9 \begin{bmatrix} \xi \\ \eta \\ \zeta \end{bmatrix} \right) \right) \right) \tag{A-11}$$

A4. Potential Energy

Define

$$\lambda_0 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

Then the potential energy of the component weights is given by

$$U_1 = \Bigg\{ M_v \lambda_0 \left[\lambda_1 + \lambda_2 (\lambda_3 + \lambda_4)\right] + M_s \lambda_0 \left[\lambda_1 + \lambda_2 \left[\lambda_3 + \lambda_{10} + \lambda_{11} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} \right]\right]$$

$$+ M_t \lambda_0 \left[\lambda_1 + \lambda_2 \left[\lambda_3 + \lambda_4 + \lambda_5 \left[\lambda_6 + \lambda_7 \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \right]\right]\right]$$

$$+ M_e \lambda_0 \left[\lambda_1 + \lambda_2 \left[\lambda_3 + \lambda_4 + \lambda_5 \left[\lambda_6 + \lambda_7 \left[\lambda_8 + \lambda_9 \begin{bmatrix} \xi_1 \\ \eta_1 \\ \zeta_1 \end{bmatrix} \right]\right]\right]\right] \qquad \text{(A-12)}$$

$$+ M_r \lambda_0 \left[\lambda_1 + \lambda_2 \left[\lambda_3 + \lambda_4 + \lambda_5 \left[\lambda_6 + \lambda_7 \left[\lambda_8 + \lambda_9 \begin{bmatrix} \xi \\ \eta \\ \zeta \end{bmatrix} \right]\right]\right]\right] \Bigg\} g$$

Assume vertical ground springs at the front and rear roller wheels. In $O_4 - X''Y''Z''$, define the coordinates of the point of contact between ground and roller wheels as l_{ij}, m_{ij}, n_{ij} for i = 1,2 and j = 1,2 where i = 1 is right, i = 2 is left, j = 1 is front and j = 2 is rear.

The coordinates of the roller wheels ground contact in O_1 - ABC are

$$\lambda_1 + \lambda_2 \left[\lambda_3 + \lambda_4 + \lambda_5 \begin{bmatrix} l_{ij} \\ m_{ij} \\ n_{ij} \end{bmatrix} \right]$$

and the extension/contraction of the springs is

$$\lambda_0 \left[\left[\lambda_1 + \lambda_2 \left[\lambda_3 + \lambda_4 + \lambda_5 \begin{bmatrix} l_{ij} \\ m_{ij} \\ n_{ij} \end{bmatrix} \right] \right] - \left[\lambda_3 + \begin{bmatrix} l_{ij} \\ m_{ij} \\ n_{ij} \end{bmatrix} \right] \right]$$

and the energy stored in the springs is

$$U_2 = \tfrac{1}{2}\,\Sigma K_{ij}\left\{\lambda_0\left[\left[\lambda_1 + \lambda_2\left[\lambda_3 + \lambda_4 + \lambda_5\begin{bmatrix} l_{ij} \\ m_{ij} \\ n_{ij}\end{bmatrix}\right]\right]\right.\right.$$

$$\left.- \left[\lambda_3 + \begin{bmatrix} l_{ij} \\ m_{ij} \\ n_{ij}\end{bmatrix}\right]\right]\Bigg\}^2 = \tfrac{1}{2}\,\Sigma K_{ij}\,\delta_{ij}^2$$

$$\text{if } \delta_{ij} < 0 \text{ then } K_{ij} = 0$$

The coordinates of the end points of the braces prior to motion are (in O_7 - U'V'W')

$$\begin{bmatrix} \underline{A_1}, \overline{A}_1 \\ \underline{A_2} \\ \underline{A_3} \end{bmatrix} - \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix}, \quad \begin{bmatrix} \pm D_1' \\ D_2' \\ D_3' \end{bmatrix}$$

Thus the original length of the brace, L_1, is

$$\left(\left(\begin{bmatrix} \underline{A_1}, \overline{A}_1 \\ \underline{A_2} \\ \underline{A_3} \end{bmatrix} - \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} - \begin{bmatrix} \pm D_1' \\ D_2' \\ D_3' \end{bmatrix}\right)^2\right)^{\frac{1}{2}} = L_1$$

The coordinates of the end points during motion (in O_7 - U'V'W') are

$$\lambda_{11}^{-1}\left(\lambda_4 + \lambda_5 \cdot \begin{bmatrix} \underline{A_1}, \overline{A}_1 \\ \underline{A_2} \\ \underline{A_3} \end{bmatrix} - \lambda_{10}\right), \quad \begin{bmatrix} \pm D_1' \\ D_2' \\ D_3' \end{bmatrix}$$

Thus the length of the brace, L_2, is

$$\left(\left(\lambda_{11}^{-1}\left(\lambda_4 + \lambda_5 \begin{bmatrix} \underline{A_1}, \overline{A}_1 \\ \underline{A_2} \\ \underline{A_3} \end{bmatrix} - \lambda_{10}\right) - \begin{bmatrix} \pm D_1' \\ D_2' \\ D_3' \end{bmatrix}\right)^2\right)^{\frac{1}{2}} = L_2$$

The energy stored in the braces is

$$U_3 = \tfrac{1}{2} K_i \left(L_1 - L_2\right)_i^2 \qquad i = 1,2 \qquad \text{(A-13)}$$

The coordinates of the attachment of the two ground springs (always perpendicular to the spade) in O_2 - ABC are ($\pm\ \alpha_1$, α_2, α_3). In the O_7 - U'V'W' system the coordinates are

$$\lambda_{11}^{-1}\left(\lambda_2^{-1}\left(\begin{bmatrix} \pm\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} - \lambda_1\right) - \lambda_3 - \lambda_{10}\right)$$

Prior to motion, the coordinates of the spring attachments in O_7 - U'V'W' are

$$\begin{bmatrix} \pm\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} - \begin{bmatrix} a_1 + g_1 \\ a_2 + g_2 \\ a_3 + g_3 \end{bmatrix}$$

The displacement of the springs, δ_i, is the "y" component in

$$\lambda_{11}^{-1}\left(\lambda_2^{-1}\left(\begin{bmatrix} \pm\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} - \lambda_1\right) - \lambda_3 - \lambda_{10}\right) - \left(\begin{bmatrix} \pm\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} - \begin{bmatrix} a_1 + g_1 \\ a_2 + g_2 \\ a_3 + g_3 \end{bmatrix}\right)$$

and so

$$U_4 = \tfrac{1}{2} K_i' \,\bar{\delta}_i^2 \qquad i = 1,2 \qquad \text{(A-14)}$$

The energy associated with the tipping parts is defined as

$$\frac{\partial U_5}{\partial q_i} = \beta_E - \frac{F(\gamma)}{L} \left[N_2 \left[- N_3 - (0_2 - d_2) \text{ Sin } \gamma + (0_3 - d_3) \text{ Cos } \gamma \right] \right.$$

$$\left. - N_3 \left[- N_2 + (0_2 - d_2) \text{ Cos } \gamma + (0_3 - d_3) \text{ Sin } \gamma \right] \right]$$

$$L = \left\{ \left[\begin{bmatrix} 0 \\ N_2 \\ N_3 \end{bmatrix} - \gamma_9^{-1} \left(\begin{bmatrix} 0 \\ 0_2 \\ 0_3 \end{bmatrix} - \lambda_8 \right) \right]^2 \right\}^{1/2}$$

where

$\beta_E = B(t) \cdot \zeta_B$ until $B(t) \cdot \zeta > \beta_{EMAX}$ then $\beta_E = \beta_{EMAX}$

when $\dot{\gamma} < 0$, $\beta_E = 0$

Also, the energy associated with the traversing parts is defined as

$$U_6 = \beta_\tau \cdot \tau$$

where

$B_\tau = B(t)\ (-\zeta_B)$ until $B(t) \mid \zeta_B \mid > \beta_{\tau MAX}$

then $B_\tau = \beta_{\tau MAX}$ when $\dot{\tau} < 0$, $\beta_\tau = 0$

The potential energy function, PE, is defined as

$$PE = \sum_1^6 U_i \qquad \text{(A-15)}$$

A5. Dissipative Energy

The dissipative function associated with the roller wheel springs is

$$\overline{U}_1 = \tfrac{1}{2} C_{ij} \left\{ \frac{d}{dt} \left\{ \lambda_0 \left[\lambda_1 + \lambda_2 \left[\lambda_3 + \lambda_4 + \lambda_5 \begin{bmatrix} l_{ij} \\ m_{ij} \\ n_{ij} \end{bmatrix} \right] \right] \right\} \right\}^2 \quad \text{(A-16)}$$

The dissipative function associated with the brace is

$$\overline{U}_2 = \tfrac{1}{2} C_{iB} \left[\frac{d}{dt} (L_1 - L_2) \right]^2$$

or since $\dot{L}_1 = 0$

$$\overline{U}_2 = \tfrac{1}{2} C_{iB} (\dot{L}_z)^2 \quad \text{(A-17)}$$

The coordinates of the end points of the spade cylinders during motion (in O_7 - U'V'W') are

$$\lambda_{11}^{-1} \left(\lambda_4 + \lambda_5 \begin{bmatrix} B_1, \overline{B}_1 \\ B_2 \\ B_3 \end{bmatrix} - \lambda_{10} \right), \begin{bmatrix} \pm C_1' \\ C_2' \\ C_3' \end{bmatrix}$$

Thus the length of the cylinder is

$$L_3 = \left(\left(\lambda_{11}^{-1} \left(\lambda_4 + \lambda_5 \begin{bmatrix} B_1, \overline{B}_1 \\ B_2 \\ B_3 \end{bmatrix} - \lambda_{10} \right) - \begin{bmatrix} \pm C_1' \\ C_2' \\ C_3' \end{bmatrix} \right)^2 \right)^{\frac{1}{2}}$$

Thus

$$\overline{U}_3 = 1/3\ \beta\ (\dot{L}_3)^3 \quad \text{(A-18)}$$

The dissipative energy function, DE, is defined as

$$\text{D.E.} = \sum_1^3 \overline{U}_i$$

Note that there is no dissipative function associated with the yaw and pitch springs of the traversing and elevating parts, respectively. Also

$$\beta = \frac{\sigma}{2g} \cdot \frac{A^3}{A_0^2}$$

where σ = specific weight of oil, A = area of piston, and A_0 = effective are of orifice.

A6. Generalized Forces

The breech force acts at (ξ_B, η_B, ζ_B) in the $0_6 - \Xi'H'Z'$ coordinate system and in the $-\eta$ direction. The recoil force, R(t), acts at (ξ_A, η_A, ζ_A) in the $0_6 - \Xi'H'Z'$ system and in the η direction but is an external force in the η equation only. The components of the breach force in 0_1 - ABC are

$$\begin{bmatrix} B(t)_A \\ B(t)_B \\ B(t)_C \end{bmatrix} = \lambda_2 \lambda_5 \lambda_7 \lambda_9 \begin{bmatrix} 0 \\ -B(t) \\ 0 \end{bmatrix}$$

In 0_1 - ABC the coordinates of the points of application are A_B, B_B, C_B

$$\begin{bmatrix} A_B \\ B_B \\ C_B \end{bmatrix} = \lambda_1 + \lambda_2 \left(\lambda_3 + \lambda_4 + \lambda_5 \left(\lambda_6 + \lambda_7 \left(\lambda_8 + \lambda_9 \begin{bmatrix} \xi_B \\ \eta_B \\ \zeta_B \end{bmatrix} \right) \right) \right)$$

The generalized force is

$$Q'_{q_i} = \begin{bmatrix} B(t)_A \\ B(t)_B \\ B(t)_C \end{bmatrix} \cdot \frac{\partial}{\partial q_i} \begin{bmatrix} A_B \\ B_B \\ C_B \end{bmatrix} \quad , \text{ where } \eta_B = \eta + \varepsilon_B$$

Similarly,

$$Q''_{\eta} = \begin{bmatrix} R(t)_A \\ R(t)_B \\ R(t)_C \end{bmatrix} \cdot \frac{\partial}{\partial q_{\eta}} \begin{bmatrix} A_R \\ B_R \\ C_R \end{bmatrix}, \text{ where } \eta_R = \eta + \varepsilon_R$$

and

$$\begin{bmatrix} R(t)_A \\ R(t)_B \\ R(t)_C \end{bmatrix} = \lambda_2 \cdot \lambda_5 \cdot \lambda_7 \cdot \lambda_9 \begin{bmatrix} 0 \\ R(t) \\ 0 \end{bmatrix}$$

and

$$\begin{bmatrix} A_R \\ B_R \\ C_R \end{bmatrix} = \lambda_1 + \lambda_2 \left(\lambda_3 + \lambda_4 + \lambda_5 \left(\lambda_6 + \lambda_7 \left(\lambda_8 + \lambda_9 \begin{bmatrix} \xi_R \\ \eta_R \\ \zeta_R \end{bmatrix} \right) \right) \right)$$

Also, a recuperator force, $C(t)$, acts at $\xi_c \ \eta_c \ \zeta_c$. Thus

$$Q'''_{\eta} = \begin{bmatrix} C(t)_A \\ C(t)_B \\ C(t)_C \end{bmatrix} \cdot \frac{\partial}{\partial q_{\eta}} \begin{bmatrix} A_c \\ B_c \\ C_c \end{bmatrix} \quad \text{where } \eta_c = \eta + \varepsilon_c$$

where

$$\begin{bmatrix} C(t)_A \\ C(t)_B \\ C(t)_C \end{bmatrix} = \lambda_2 \cdot \lambda_5 \cdot \lambda_7 \cdot \lambda_8 \begin{bmatrix} 0 \\ C(t) \\ 0 \end{bmatrix}$$

and

$$\begin{bmatrix} A_c \\ B_c \\ C_c \end{bmatrix} = \lambda_1 + \lambda_2 \left(\lambda_3 + \lambda_4 + \lambda_5 \left(\lambda_6 + \lambda_7 \left(\lambda_8 + \lambda_9 \begin{bmatrix} \xi_c \\ \eta_c \\ \zeta_c \end{bmatrix} \right) \right) \right)$$

Define

$$\lambda_0' = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

then the displacement of the horizontal ground springs is

$$\delta'_{ij} = \lambda_0' \left[\left[\lambda_1 + \lambda_2 \left[\lambda_3 + \lambda_4 + \lambda_5 \begin{bmatrix} l_{ij} \\ m_{ij} \\ n_{ij} \end{bmatrix} \right] \right] \right.$$

$$\left. - \left[\lambda_3 + \begin{bmatrix} l_{ij} \\ m_{ij} \\ n_{ij} \end{bmatrix} \right] \right]$$

and the horizontal force is $- K'_{ij}\ \lambda'_{ij}$. The generalized force is Q^{iv}_{ij} such that

$$F_{ij} = \begin{cases} -K'_{ij}\delta'_{ij} & \text{if} \quad \left| - K'_{ij}\ \delta'_{ij} \right| < \mu\ K_{ij}\ \delta_{ij} \\ -\mu\ K_{ij}\ \delta_{ij}\ (\text{signum}\ \delta'_{ij}) & \text{if} \quad \left| - K'_{ij}\ \delta'_{ij} \right) \geq \mu\ K_{ij}\ \delta_{ij} \end{cases}$$

Note when $\delta_{ij} \leq 0$, $K_{ij} = K'_{ij} = 0$

$$Q^{iv}_{ij} = \begin{bmatrix} F_{ij} \\ 0 \\ 0 \end{bmatrix} \cdot \frac{\partial}{\partial q_i} \cdot \left[\lambda_1 + \lambda_2 \left(\lambda_3 + \lambda_4 + \lambda_5 \begin{bmatrix} l_{ij} \\ m_{ij} \\ n_{ij} \end{bmatrix} \right) \right]$$

APPENDIX B

Mathematics of Solution Technique

APPENDIX B

Mathematics of Solution Technique

Since FORMAC takes only partial derivatives the following analysis was required to determine which derivatives are actually needed and once they are obtained in what manner are they combined to produce the equations of motion.

To begin the analysis the expression for the Lagrangian which yields the equations of motion is

$$\frac{d}{dt}\frac{\partial(KE)}{\partial\dot{q}_j} - \frac{\partial(KE)}{\partial q_j} + \frac{\partial(DE)}{\partial\dot{q}_j} + \frac{\partial(PE)}{\partial q_j} = F_j \qquad \text{(B-1)}$$

where $j = 1, 2, \ldots, k$

KE = Total kinetic energy

DE = Total dissipative energy

PE = Total potential energy

F_j = Generalized Force

q_j = Generalized coordinate

$\dot{q}_j$ = Generalized velocity

t = Independent variable, time

k = Number of generalized coordinates

Consider at first only the kinetic energy terms of equation (B-1) and write

$$\frac{d}{dt}\frac{\partial(KE)}{\partial\dot{q}_j} - \frac{\partial(KE)}{\partial q_j} = \frac{d}{dt}\frac{\partial(T+W)}{\partial\dot{q}_j} - \frac{\partial(T+W)}{\partial q_j} \qquad \text{(B-2)}$$

where $T = \sum_{i=1}^{n} T_i$, T is the translational part of the kinetic energy

$W = \sum_{i=1}^{n} W_i$, W is the rotational part of the kinetic energy

n = Number of masses

$T_i = 1/2\ M_i \dot{\vec{Q}}_i \cdot \dot{\vec{Q}}_i$

$W_i = \dot{\vec{Q}}_i \cdot \vec{\omega}_i \int_{M_i} \rho_i dM_i + 1/2 \int_{M_i} (\vec{\omega}_i \times \vec{\rho}_i) \cdot (\vec{\omega}_i \times \vec{\rho}_i) dM_i$

M_i = $i^{\underline{th}}$ mass

$\vec{Q}_i$ = Position vector of the $i^{\underline{th}}$ mass

$\vec{\rho}_i$ = Vector from the mass center of M_i to an element of M_i

ω_i = Angular velocity of mass M_i

Now consider only the T_i and dropping the subscript on the T for convenience as though only one mass is being studied at this time. An examination of the expression (B-3) will proceed.

$$\frac{d}{dt} \frac{\partial \dot{\vec{Q}} \cdot \dot{\vec{Q}}}{\partial \dot{q}} \tag{B-3}$$

$$\frac{\partial \dot{\vec{Q}} \cdot \dot{\vec{Q}}}{\partial \dot{q}} = \dot{\vec{Q}}_{\dot{q}} \cdot \dot{\vec{Q}} + \dot{\vec{Q}} \cdot \dot{\vec{Q}}_{\dot{q}} = 2\dot{\vec{Q}}_{\dot{q}} \cdot \dot{\vec{Q}}$$

where $\vec{Q}_q = \frac{\partial \vec{Q}}{\partial \mathbf{q}}$; $q = [q_1, q_2, \ldots, q_k]^T$

Since Q is a function of the generalized coordinates only, i.e.,

$$\vec{Q} = \vec{Q}(q_1, q_2, \ldots, q_k)$$

then $$\frac{d}{dt}\vec{Q} = \dot{\vec{Q}} = \frac{\partial \vec{Q}}{\partial q_1}\dot{q}_1 + \frac{\partial \vec{Q}}{\partial q_2}\dot{q}_2 + \ldots + \frac{\partial \vec{Q}}{\partial q_k}\dot{q}_k = \vec{Q}_{q_i}\dot{q}_i \tag{B-4}$$

where the double subscript on i refers to the Einstein summation notation and runs to the value k. From (B-4)

$$\dot{\vec{Q}}_{\dot{q}} = [\vec{Q}_{q_1}\dot{q}_1 + \vec{Q}_{q_2}\dot{q}_2 + \dots + \vec{Q}_{q_k}\dot{q}_k]_{\dot{q}}$$

$$= \vec{Q}_{q_1\dot{q}}\dot{q}_1 + \vec{Q}_{q_1}\dot{q}_{1\dot{q}} + \vec{Q}_{q_2\dot{q}}\dot{q}_2 + \vec{Q}_{q_2}\dot{q}_{2\dot{q}} + \dots + \vec{Q}_{q_k\dot{q}}\dot{q}_k + \vec{Q}_{q_k}\dot{q}_{k\dot{q}}$$

Since $\vec{Q}$ is a function of only the coordinates

$$\dot{\vec{Q}}_{\dot{q}} = \frac{\partial\vec{Q}}{\partial q_1}\frac{\partial\dot{q}_1}{\partial\dot{q}} + \frac{\partial\vec{Q}}{\partial q_2}\frac{\partial\dot{q}_2}{\partial\dot{q}} + \dots + \frac{\partial\vec{Q}}{\partial q_k}\frac{\partial\dot{q}_k}{\partial\dot{q}} = \frac{\partial\vec{Q}}{\partial q}$$

or $$\dot{\vec{Q}}_{\dot{q}} = \vec{Q}_q \qquad \text{(B-5)}$$

Therefore,

$$\frac{\partial\dot{\vec{Q}}\cdot\dot{\vec{Q}}}{\partial\dot{q}} = 2\dot{\vec{Q}}_{\dot{q}}\cdot\dot{\vec{Q}} = 2\vec{Q}_q\cdot\dot{\vec{Q}}$$

and

$$\frac{d}{dt}\left[\frac{\partial\dot{\vec{Q}}\cdot\dot{\vec{Q}}}{\partial\dot{q}}\right] = \frac{d}{dt}\left[2\vec{Q}_q\cdot\dot{\vec{Q}}\right]$$

For one mass, say $T = 1/2M\dot{\vec{Q}}\cdot\dot{\vec{Q}}$, then

$$\frac{d}{dt}\left[\frac{\partial T}{\partial\dot{q}}\right] = \frac{d}{dt}\left[\frac{\partial 1/2\, M\dot{\vec{Q}}\cdot\dot{\vec{Q}}}{\partial\dot{q}}\right] = \frac{d}{dt}\left[M\vec{Q}_q\cdot\dot{\vec{Q}}\right]$$

Examination of the term

$$\frac{d}{dt}\ [\vec{Q}_q \cdot \dot{\vec{Q}}]$$

yields

$$\frac{d}{dt}\ [\vec{Q}_q \cdot \dot{\vec{Q}}] = [\vec{Q}_q \cdot \dot{\vec{Q}}]_{q_1} \dot{q}_1 + [\vec{Q}_q \cdot \dot{\vec{Q}}]_{q_2} \dot{q}_2 + \ldots + [\vec{Q}_q \cdot \dot{\vec{Q}}]_{q_k} \dot{q}_k$$

$$+ [\vec{Q}_q \cdot \dot{\vec{Q}}]_{\dot{q}_1} \ddot{q}_1 + [\vec{Q}_q \cdot \dot{\vec{Q}}]_{\dot{q}_2} \ddot{q}_2 + \ldots + [\vec{Q}_q \cdot \dot{\vec{Q}}]_{\dot{q}_k} \ddot{q}_k$$

$$\frac{d}{dt}\ [\vec{Q}_q \cdot \dot{\vec{Q}}] = [\vec{Q}_{qq_1} \cdot \dot{\vec{Q}} + \vec{Q}_q \cdot \dot{\vec{Q}}_{q_1}] \dot{q}_1 + \ldots + [\vec{Q}_{qq_k} \cdot \dot{\vec{Q}} + Q_q \cdot \dot{\vec{Q}}_{q_k}] \dot{q}_k$$

$$+ [\vec{Q}_{q\dot{q}_1} \cdot \dot{\vec{Q}} + \vec{Q}_q \cdot \dot{\vec{Q}}_{\dot{q}_1}] \ddot{q}_1 + \ldots + [\vec{Q}_{q\dot{q}_k} \cdot \dot{\vec{Q}} + \vec{Q}_q \cdot \dot{\vec{Q}}_{\dot{q}_k}] \ddot{q}_k$$

Since $\vec{Q}$ is a function of only the coordinates, terms like $Q_{q\dot{q}_i}$ go to zero and using the results of equation (B-5) the above expression reduces to

$$\frac{d}{dt}\ [\vec{Q}_q \cdot \dot{\vec{Q}}] = [\vec{Q}_{qq_1} \cdot \dot{\vec{Q}} + \vec{Q}_q \cdot \dot{\vec{Q}}_{q_1}] \dot{q}_1 + \ldots + [\vec{Q}_{qq_k} \cdot \dot{\vec{Q}} + \vec{Q}_q \cdot \dot{\vec{Q}}_{q_k}] \dot{q}_k$$

$$+ \vec{Q}_q \cdot \vec{Q}_{q_1} \ddot{q}_1 + \ldots + \vec{Q}_q \cdot \vec{Q}_{q_k} \ddot{q}_k \tag{B-6}$$

Since

$$\dot{\vec{Q}}_q = \left[\frac{\partial \vec{Q}}{\partial q_1}\dot{q}_1 + \frac{\partial \vec{Q}}{\partial q_2}\dot{q}_2 + \ldots + \frac{\partial \vec{Q}}{\partial q_k}\dot{q}_k\right]_q$$

$$= \left[\frac{\partial^2 \vec{Q}}{\partial q_1 \partial q}\dot{q}_1 + \frac{\partial^2 \vec{Q}}{\partial q_2 \partial q}\dot{q}_2 + \ldots + \frac{\partial^2 \vec{Q}}{\partial q_k \partial q}\dot{q}_k\right] = \vec{Q}_{q_i q}\dot{q}_i \qquad \text{(B-7)}$$

And using the results of equations B-4, 7, equation B-6 becomes

$$\frac{d}{dt}\left[\vec{Q}_q \cdot \dot{\vec{Q}}\right] = \left[\vec{Q}_{qq_1} \cdot \vec{Q}_{q_i}\dot{q}_i + \vec{Q}_q \cdot \vec{Q}_{q_i q_1}\dot{q}_i\right]\dot{q}_1 + \ldots + \left[\vec{Q}_{qq_k} \cdot \vec{Q}_{q_i}\dot{q}_i + \right.$$

$$\left. \vec{Q}_q \cdot \vec{Q}_{q_i q_k}\dot{q}_i\right]\dot{q}_k + \vec{Q}_q \cdot \vec{Q}_{q_1}\ddot{q}_1 + \ldots + \vec{Q}_q \cdot \vec{Q}_{q_k}\ddot{q}_k$$

$$= \dot{q}_1 \vec{Q}_{qq_1} \cdot \vec{Q}_{q_i}\dot{q}_i + \ldots + \dot{q}_k \vec{Q}_{qq_k} \cdot \vec{Q}_{q_i}\dot{q}_i$$

$$+ \vec{Q}_q \cdot \vec{Q}_{q_i q_1}\dot{q}_i\dot{q}_1 + \ldots + \vec{Q}_q \cdot \vec{Q}_{q_i q_k}\dot{q}_i\dot{q}_k$$

$$+ \vec{Q}_q \cdot \vec{Q}_{q_1}\ddot{q}_1 + \ldots + \vec{Q}_q \cdot \vec{Q}_{q_k}\ddot{q}_k$$

$$= \left[\dot{q}_1 \vec{Q}_{qq_1} + \ldots + \dot{q}_k \vec{Q}_{qq_k}\right] \cdot \vec{Q}_{q_i}\dot{q}_i$$

$$+ \vec{Q}_q \cdot \left[\vec{Q}_{q_i q_1}\dot{q}_i\dot{q}_1 + \ldots + \vec{Q}_{q_i q_k}\dot{q}_i\dot{q}_k\right]$$

$$+ \vec{Q}_q \cdot \left[\vec{Q}_{q_1}\ddot{q}_1 + \ldots + \vec{Q}_{q_k}\ddot{q}_k\right]$$

$$= \dot{q}_j \vec{Q}_{qq_j} \cdot \vec{Q}_{q_i}\dot{q}_i + \vec{Q}_q \cdot \vec{Q}_{q_i q_j}\dot{q}_i\dot{q}_j + \vec{Q}_q \cdot \vec{Q}_{q_i}\ddot{q}_i$$

Thus, for one mass the following result is obtained

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{q}} = \frac{d}{dt}\left[M\vec{Q}_q\cdot\dot{\vec{Q}}\right] = M\left[\dot{q}_j\vec{Q}_{qq_j}\cdot\vec{Q}_{q_i}\dot{q}_i + \vec{Q}_q\cdot\vec{Q}_{q_iq_j}\dot{q}_i\dot{q}_j + \vec{Q}_q\cdot\vec{Q}_{q_i}\ddot{q}_i\right] \tag{B-8}$$

When more than one mass is involved a summation over all masses must be performed. Equation (B-8) would take the form

$$\frac{d}{dt}\frac{\partial\Sigma T_i}{\partial\dot{q}} \tag{B-9}$$

Before proceeding with the rotational part of equation (B-2), an examination of the translational part of the second term of that equation is given below. For ease of notation consider only one mass. Then,

$$\frac{\partial T}{\partial q} = \frac{\partial}{\partial q}\left[1/2\,M\dot{\vec{Q}}\cdot\dot{\vec{Q}}\right]\ ;\ q = \left[q_1,\ q_2,\ \cdots\ q_k\right]$$

and

$$\frac{\partial}{\partial q}\left[1/2\,M\dot{\vec{Q}}\cdot\dot{\vec{Q}}\right] = \frac{M}{2}\left[\dot{\vec{Q}}_q\cdot\dot{\vec{Q}} + \dot{\vec{Q}}\cdot\dot{\vec{Q}}_q\right] = \frac{M}{2}\left[2\dot{\vec{Q}}_q\cdot\dot{\vec{Q}}\right]$$

or

$$\frac{\partial T}{\partial q} = M\dot{\vec{Q}}_q\cdot\dot{\vec{Q}} = M\,Q_{q_iq}\dot{q}_i\cdot\vec{Q}_{q_j}\dot{q}_j \tag{B-10}$$

from B-4 and B-7. The results of equation (B-10) also appear in equation (B-8); and since (B-10) is preceded by a minus sign, these terms cancel and equation (B-2) becomes

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{q}} - \frac{\partial T}{\partial q} = \Sigma M \left[\vec{Q}_q \cdot \vec{Q}_{q_i q_j} \dot{q}_i \dot{q}_j + \vec{Q}_q \cdot \vec{Q}_{q_i} \ddot{q}_i\right] \qquad \text{(B-11)}$$

It was seen earlier that $W = \sum_{i=1}^{n} W_i$, where

$$W_i = \dot{\vec{Q}}_i \cdot \vec{\omega}_i \int_{M_i} \rho_i dM_i + 1/2 \int_{M_i} (\vec{\omega}_i \times \vec{\rho}_i) \cdot (\vec{\omega}_i \times \vec{\rho}_i)\, dM_i$$

$$= 1/2\, I_{xx}(i)\, \omega_x^2(i) + 1/2\, I_{yy}(i)\, \omega_y^2(i) + 1/2\, I_{zz}(i)\, \omega_z^2(i) \qquad \text{(B-12)}$$

$$- \left[I_{xy}(i)\, \omega_x(i)\, \omega_y(i) + I_{xz}(i)\, \omega_x(i)\, \omega_z(i) + I_{yz}(i)\, \omega_y(i)\, \omega_z(i)\right]$$

since $\int_{M_i} \rho_i\, dM_i = 0$. Equation (B-12) can be written in matrix form as

$W_i = A(i)\ B(i)$

Where A(i) and B(i) are defined as

$$A(i) = \begin{bmatrix} 1/2\, I_{xx}(i)\, \omega_x(i) - I_{xy}(i)\, \omega_y(i) \\ 1/2\, I_{yy}(i)\, \omega_y(i) - I_{yz}(i)\, \omega_z(i) \\ 1/2\, I_{zz}(i)\, \omega_z(i) - I_{xz}(i)\, \omega_x(i) \end{bmatrix}^T ; \quad B(i) = \begin{bmatrix} \omega_x(i) \\ \omega_y(i) \\ \omega_z(i) \end{bmatrix}$$

The subscripts are dropped again as a matter of notational convenience.

$$\frac{\partial W}{\partial \dot{q}} = W_{\dot{q}} = (AB)_{\dot{q}} = A_{\dot{q}} B + A B_{\dot{q}}$$

Since W is a function of the generalized coordinates and generalized velocities i.e., $W = W(q_1, q_2, \ldots, q_k, \dot{q}_1, \dot{q}_2, \ldots, \dot{q}_k)$ then

$$\frac{d}{dt}[W_{\dot{q}}] = A_{\dot{q}q_1} B\dot{q}_1 + A_{\dot{q}} B_{q_1} \dot{q}_1 + \cdots + A_{\dot{q}q_k} B\dot{q}_k + A_{\dot{q}} B_{q_k} \dot{q}_k$$

$$+ A_{q_1} B_{\dot{q}} \dot{q}_1 + A B_{\dot{q}q_1} \dot{q}_1 + \cdots + A_{q_k} B_{\dot{q}} \dot{q}_k + A B_{\dot{q}q_k} \dot{q}_k$$

$$+ A_{\dot{q}\dot{q}_1} B\ddot{q}_1 + A_{\dot{q}} B_{\dot{q}_1} \ddot{q}_1 + \cdots + A_{\dot{q}\dot{q}_k} B\ddot{q}_k + A_{\dot{q}} B_{\dot{q}_k} \ddot{q}_k \qquad \text{(B-13)}$$

$$+ A_{\dot{q}_1} B_{\dot{q}} \ddot{q}_1 + A B_{\dot{q}\dot{q}_1} \ddot{q}_1 + \cdots + A_{\dot{q}_k} B_{\dot{q}} \ddot{q}_k + A B_{\dot{q}\dot{q}_k} \ddot{q}_k$$

However, terms like $A_{\dot{q}\dot{q}_i}$ and $B_{\dot{q}\dot{q}_i}$ go to zero because each of the vectors is linear with respect to the generalized velocities. Equation (B-13) reduces to

$$\frac{d}{dt}\left[W_{\dot{q}}\right] = \dot{q}_i A_{\dot{q}q_i} B + A_{\dot{q}} B_{q_j} \dot{q}_j + \dot{q}_i A_{q_i} B_{\dot{q}} + AB_{\dot{q}q_i} \dot{q}_i$$

$$+ A_{\dot{q}} B_{q_i} \ddot{q}_i + \ddot{q}_j A_{\dot{q}_j} B_{\dot{q}} \qquad \text{(B-14)}$$

The remainder of the rotational terms are

$$\frac{\partial W}{\partial q} = A_q B + A B_q \qquad \text{(B-15)}$$

The above analysis describes which partial derivatives must be taken and how they are combined for the kinetic energy. The derivatives of the other energy expressions are either obtained from the kinetic energy or are completed separately. For the most part, those differential expressions which cannot be obtained from the kinetic energy are easily calculated by the use of FORMAC. In order for the reader to see more clearly the use of equations (B-11), (B-14), and (B-15), a discussion is

given in section 2.0 with FORTRAN to see exactly how terms are combined. Also, in section 3.3 this is discussed further with actual application to an artillery problem.

APPENDIX C

FORMAC Program

APPENDIX C

FORMAC Program

This appendix contains a listing of the FORMAC program which was used to generate the differential expressions to solve Lagrange's Equation. This program was developed on the IBM 360/65 operating system Time Sharing Option (TSO). A maximum of 230K bytes of core storage was available and thus segments of the program were commented out when not utilized. Because of the ease in which TSO converses with the user, the values of some variables were changed to accommodate a quick formulation for the next energy expression to be evaluated. At all times maximum use was made of all previous coding and so some expressions may appear, at first glance, out of place.

```
START: PROCEDURE OPTIONS(MAIN);
         DCL LINE CHAR(600) VAR;
         FORMAC_OPTIONS;
         OPTSET(NOEDIT);
         OPTSET(LINELENGTH=80);
     CARDS =0;
     PTNO=6;  /*          PT(I,J)---PTNO IS I PT NUMBER                    */
/*                                                                         */
     LET(CHAN1=CHAIN(COS(Q(6)),C6,SIN(Q(6)),S6,COS(Q(7)),C7,SIN(Q(7)),
                     S7,COS(Q(8)),C8,SIN(Q(8)),S8,COS(Q(9)),C9,
                     SIN(Q(9)),S9,COS(Q(10)),C10,SIN(Q(10)),S10,
                      COS(Q(11)),C11,SIN(Q(11)),S11));
/*                                                                         */
     /*
     LET(CHAN2=CHAIN(C8*S9 + S8*C9*S10,ZZ(1),-C8*S9-S8*C9*S10,ZZ(2),
         C8*C9*S10-S8*S9,ZZ(3),-C8*C9*S10 + S8*S9,ZZ(4),C8*S9*S10 + S8*
         C9,ZZ(5),-C8*S9*S10-S8*C9,ZZ(6),C8*C9-S8*S9*S10,ZZ(7),
         -C8*C9 + S8*S9*S10,ZZ(8)));
     */
     LET(CHAN3=CHAIN(C9*S10*S6 + S9*C6,ZZ(9),-C9*S10*S6-S9*C6,ZZ(10),
         C9*S10*C6-S9*S6,ZZ(11),-C9*S10*C6 + S9*S6,ZZ(12),
         C9*S6 + S9*S10*C6,ZZ(13),-C9*S6-S9*S10*C6,ZZ(14),
         C9*C6-S9*S10*S6,ZZ(15),-C9*C6 + S9*S10*S6,ZZ(16)));
     /*
     LET(CHAN7=CHAIN(A2*C6-A3*S6,ZZ(41),A2*S6 + A3*C6,ZZ(42))),
     LET(CHAN8=CHAIN(-G3 + Q(5) + ZZ(42),ZZ(43),
         -G2-Q(2) + Q(4) + ZZ(41),ZZ(44)));
     LET(CHAN9=CHAIN(-D2P + ZZ(43)*S8 + ZZ(44)*C8,ZZ(45),
         -D3P + ZZ(43)*C8-ZZ(44)*S8,ZZ(46)));
     LET(CHAN10=CHAIN(D1P + A1BAR-G1,ZZ(54),-G2-D2P + A2,ZZ(55),
         -D3P + A3-G3,ZZ(49)));
     LET(CHAN11=CHAIN(ZZ(54)**2 + ZZ(45)**2 + ZZ(46)**2,ZZ(56),
         ZZ(54)**2 + ZZ(55)**2 + ZZ(49)**2,ZZ(57))),
     LET(CHAN12=CHAIN(-ZZ(56)** 5 + ZZ(57)** 5,ZZ(58))),
     LET(CHAN13=CHAIN(XKK2*ZZ(58)*ZZ(56)**(- 5 ),ZZ(59))),
     */
     /*
```

```
      LET(CHAN10=CHAIN(-D1P-G1 + A1,ZZ(47),-D2P-G2 + A2,ZZ(48),
          -D3P + A3-G3,ZZ(49)));
      LET(CHAN11=CHAIN(ZZ(47)**2 + ZZ(45)**2 + ZZ(46)**2,ZZ(50),
          ZZ(47)**2 + ZZ(48)**2 + ZZ(49)**2,ZZ(51)));
      LET(CHAN12=CHAIN(-ZZ(50)**.5 + ZZ(51)**.5,ZZ(52)));
      LET(CHAN13=CHAIN(ZZ(52)*XKK1*ZZ(50)**(-.5 ),ZZ(53)));
      */
      /*
      LET(CHAN14=CHAIN(-XN(1,1)-A3 + XN(1,1)*ZZ(15)-XL(1,1)*S9*C10 + XM(
1,1)*ZZ(13) + A2*S9*S10 + A3*C9 + Q(4)*S9*S10 + Q(5)*C9-A1*S9*C10,
ZZ(60)));
      LET(CHAN15=CHAIN(-XN(1,2)-A3-XL(1,2)*S9*C10 + XM(1,2)*ZZ(13) + XN(
1,2)*ZZ(15) + Q(4)*S9*S10 + Q(5)*C9-A1*S9*C10 + A3*C9 + A2*S9*S10,
ZZ(61)));
      LET(CHAN16=CHAIN(-XN(2,1)-A3-XL(2,1)*S9*C10 + XM(2,1)*ZZ(13) + XN(
2,1)*ZZ(15) + Q(4)*S9*S10 + Q(5)*C9-A1*S9*C10 + A3*C9 + A2*S9*S10,
ZZ(62)));
       LET(CHAN17=CHAIN(-XN(2,2)-A3 + XM(2,2)*ZZ(13) + XN(2,2)*ZZ(15)-XL
(2,2)*S9*C10 + Q(4)*S9*S10 + Q(5)*C9-A1*S9*C10 + A3*C9 + A2*S9*S10,
ZZ(63)));
      */
      /*
      LET(CHAN18=CHAIN(-ALPHA2 + G2 + Q(2) + A2 + C8*(-G2-Q(2)-A2 + ALPH
A2*C10 + ALPHA3*S9*S10-C9*S10*(ALPHA1-Q(3))) + S8*(-G3-A3 + ALPHA3*C9 +
 S9*(ALPHA1-Q(3))),ZZ(64)));
      */
      /*
      LET(CHAN18=CHAIN(-ALPHA2 + G2 + Q(2) + A2 + C8*(-G2-Q(2)-A2 + ALPH
A2*C10 + ALPHA3*S9*S10-C9*S10*(-ALPHA1-Q(3))) + S8*(-A3-G3 + ALPHA3*C9
+ S9*(-ALPHA1-Q(3))),ZZ(65)));
      */
      /*
      LET(CHAN19=CHAIN(-G2-Q(2) + Q(4) + B2*C6-B3*S6,ZZ(66))),
      LET(CHAN20=CHAIN(-G3 + Q(5) + B2*S6 + B3*C6,ZZ(67),
          -C1P + B1-G1,ZZ(68)));
      LET(CHAN21=CHAIN(-C2P + ZZ(67)*S8 + ZZ(66)*C8,ZZ(69),
          -C3P + ZZ(67)*C8-ZZ(66)*S8,ZZ(70))).
```

```
      LET(CHAN22=CHAIN((ZZ(70)**2 + ZZ(68)**2 + ZZ(69)**2)**(-.5 )*.5,
          ZZ(71),-B2*S6-B3*C6,ZZ(72),B2*C6-B3*S6,ZZ(73)));
      LET(CHAN23=CHAIN((ZZ(70)**2 + ZZ(69)**2 + (B1BAR + C1P-G1)**2)**(-
.5 )*.5,ZZ(74)));
      */
      /*
      LET(CHAN24=CHAIN(-G2-Q(2) + Q(4)-A3*S6 + A2*C6,ZZ(75),
          -G3 + Q(5) + A3*C6 + A2*S6,ZZ(76)));
      LET(CHAN25=CHAIN(-D2P + ZZ(76)*S8 + ZZ(75)*C8,ZZ(77),
          -D3P + ZZ(76)*C8-ZZ(75)*S8,ZZ(78),-A3*S6 + A2*C6,ZZ(79),
          -A3*C6-A2*S6,ZZ(80)));
      LET(CHAN26=CHAIN((ZZ(77)**2 + ZZ(78)**2 + (D1P + A1BAR-G1)**2)**(-
.5 )*.5,ZZ(81)));
      LET(CHAN27=CHAIN((ZZ(77)**2 + ZZ(78)**2 + (A1-D1P-G1)**2)**(-.5 )*
.5,ZZ(82)));
      */
      /*
      LET(CHAN28=CHAIN(A2*S9*S10 + A3*C9 + Q(4)*S9*S10 + Q(5)*C9-A1*S9*C
10,ZZ(83)));
      LET(CHAN29=CHAIN(-A2*C9*S10 + A3*S9-Q(4)*C9*S10 + Q(5)*S9 + A1*C9*
C10,ZZ(84)));
      */
      LET(CHAN30=CHAIN(ZZ(12)*C11-C9*C10*S11,ZZ(85),
          -C9*C10*C11*C6 + C9*S10*S11,ZZ(86),C10*C11*C6-S10*S11,ZZ(87),
          -C10*S11*C6-S10*C11,ZZ(88),S9*C10*C11*C6-S9*S10*S11,ZZ(89),
          ZZ(13)*C11 + S9*C10*S11,ZZ(90)));
/*                                                                      */
      LET(LAM1(1,1)=Q(3); LAM1(2,1)=0; LAM1(3,1)=0);
      LET(LAMA(1,1)=COS(Q(9)); LAMA(1,2)=0; LAMA(1,3)=SIN(Q(9)));
      LET(LAMA(2,1)=0; LAMA(2,2)=1; LAMA(2,3)=0);
      LET(LAMA(3,1)=-SIN(Q(9)); LAMA(3,2)=0, LAMA(3,3)=COS(Q(9)));
      LET(LAMB(1,1)=COS(Q(10)); LAMB(1,2)=-SIN(Q(10)); LAMB(1,3)=0);
      LET(LAMB(2,1)=SIN(Q(10)); LAMB(2,2)=COS(Q(10)); LAMB(2,3)=0),
      LET(LAMB(3,1)=0; LAMB(3,2)=0; LAMB(3,3)=1);
      LET(LAM3(1,1)=A1, LAM3(2,1)=A2, LAM3(3,1)=A3),
      LET(LAM4(1,1)=0; LAM4(2,1)=Q(4), LAM4(3,1)=Q(5)).
      LET(LAM5(1,1)=1, LAM5(1,2)=0, LAM5(1,3)=0),
```

```
        LET(LAM5(2,1)=0; LAM5(2,2)=COS(Q(6)); LAM5(2,3)=-SIN(Q(6)));
        LET(LAM5(3,1)=0; LAM5(3,2)=SIN(Q(6)); LAM5(3,3)=COS(Q(6)));
        LET(LAM6(1,1)=E1; LAM6(2,1)=E2; LAM6(3,1)=E3);
        LET(LAM7(1,1)=COS(Q(11)); LAM7(1,2)=-SIN(Q(11)); LAM7(1,3)=0);
        LET(LAM7(2,1)=SIN(Q(11)); LAM7(2,2)=COS(Q(11)); LAM7(2,3)=0);
        LET(LAM7(3,1)=0; LAM7(3,2)=0; LAM7(3,3)=1);
        LET(LAM8(1,1)=D1; LAM8(2,1)=D2; LAM8(3,1)=D3);
        LET(LAM9(1,1)=1; LAM9(1,2)=0; LAM9(1,3)=0);
        LET(LAM9(2,1)=0; LAM9(2,2)=COS(Q(7)); LAM9(2,3)=-SIN(Q(7)));
        LET(LAM9(3,1)=0; LAM9(3,2)=SIN(Q(7)); LAM9(3,3)=COS(Q(7)));
        LET(LAM10(1,1)=G1; LAM10(2,1)=G2 + Q(2); LAM10(3,1)=G3);
        LET(LAM11(1,1)=1; LAM11(1,2)=0; LAM11(1,3)=0);
        LET(LAM11(2,1)=0; LAM11(2,2)=COS(Q(8)); LAM11(2,3)=-SIN(Q(8)));
        LET(LAM11(3,1)=0; LAM11(3,2)=SIN(Q(8)); LAM11(3,3)=COS(Q(8)));
        LET(H(1,1)=HH1; H(2,1)=HH2; H(3,1)=HH3);
        LET(F(1,1)=FF1; F(2,1)=FF2; F(3,1)=FF3);
        LET(E(1,1)=XI1; E(2,1)=ETA1; E(3,1)=ZETA1);
        LET(R(1,1)=XI; R(2,1)=Q(1); R(3,1)=ZETA);
/*                                                                  */
/*              MULTIPLY LAMA TIMES LAMB TO OBTAIN LAM2             */
/*                                                                  */
LOOP1: DO I=1 TO 3; LET(I="I");
LOOP2: DO J=1 TO 3; LET(J="J");
        LET(LAM2(I,J)=0);
LOOP3: DO K=1 TO 3; LET(K="K");
        LET(LAM2(I,J)=LAM2(I,J) + LAMA(I,K)*LAMB(K,J));
END LOOP3; END LOOP2; END LOOP1;
/*                                                                  */
/*              MULTIPLY LAM2*LAM11 AND LAM2*LAM5                   */
/*                                                                  */
LOOP4: DO I=1 TO 3; LET(I="I");
LOOP5: DO J=1 TO 3; LET(J="J");
        LET(LAM211(I,J)=0; LAM25(I,J)=0);
LOOP6: DO K=1 TO 3; LET(K="K");
        LET(LAM211(I,J)=LAM211(I,J) + LAM2(I,K)*LAM11(K,J));
        LET(LAM25(I,J)=LAM25(I,J) + LAM2(I,K)*LAM5(K,J));
END LOOP6; END LOOP5; END LOOP4;
```

```
/*                                                                    */
/*                    MULTIPLY LAM25*LAM7                             */
/*                                                                    */
LOOP7: DO I=1 TO 3; LET(I="I");
LOOP8: DO J=1 TO 3; LET(J="J");
     LET(LAM257(I,J)=0);
LOOP9: DO K=1 TO 3; LET(K="K");
     LET(LAM257(I,J)=LAM257(I,J) + LAM25(I,K)*LAM7(K,J));
END LOOP9; END LOOP8; END LOOP7;
/*                                                                    */
/*                    MULTIPLY LAM257*LAM9                            */
/*                                                                    */
LOOP10: DO I=1 TO 3; LET(I="I");
LOOP11: DO J=1 TO 3; LET(J="J");
      LET(LAM2579(I,J)=0);
LOOP12: DO K=1 TO 3; LET(K="K");
      LET(LAM2579(I,J)=LAM2579(I,J) + LAM257(I,K)*LAM9(K,J));
END LOOP12; END LOOP11; END LOOP10;
/*                                                                    */
/*                    NOW MULTIPLY ALL (3X3)'S BY (3X1)'S             */
/*                                                                    */
LOOP13: DO I=1 TO 3; LET(I="I");
LOOP14: DO J=1 TO 1; LET(J="J");
      LET(LAM23(I,J)=0; LAM24(I,J)=0; LAM210(I,J)=0);
      LET(LAM211H(I,J)=0; LAM256(I,J)=0; LAM257F(I,J)=0);
      LET(LAM2578(I,J)=0; LAM2579E(I,J)=0; LAM2579R(I,J)=0);
LOOP15: DO K=1 TO 3; LET(K="K");
      LET(LAM23(I,J)=LAM23(I,J) + LAM2(I,K)*LAM3(K,J));
      LET(LAM24(I,J)=LAM24(I,J) + LAM2(I,K)*LAM4(K,J));
      LET(LAM210(I,J)=LAM210(I,J) + LAM2(I,K)*LAM10(K,J));
      LET(LAM211H(I,J)=LAM211H(I,J) + LAM211(I,K)*H(K,J));
      LET(LAM256(I,J)=LAM256(I,J) + LAM25(I,K)*LAM6(K,J));
      LET(LAM257F(I,J)=LAM257F(I,J) + LAM257(I,K)*F(K,J));
      LET(LAM2578(I,J)=LAM2578(I,J) + LAM257(I,K)*LAM8(K,J));
      LET(LAM2579E(I,J)=LAM2579E(I,J) + LAM2579(I,K)*E(K,J));
      LET(LAM2579R(I,J)=LAM2579R(I,J) + LAM2579(I,K)*R(K,J));
END LOOP15; END LOOP14; END LOOP13;
```

```
/*                                                                 */
/*                    EVALUATE PT(I,J)'S                           */
/*                                                                 */
      /*
LOOP16: DO J=1 TO 3; LET(J="J");
        LET(PT(1,J)=LAM1(J,1) + LAM23(J,1));
        ATOMIZE(LAM23(J,1));
        LET(PT(2,J)=LAM24(J,1));
        ATOMIZE(LAM24(J,1));
        LET(PT(3,J)=LAM210(J,1) + LAM211H(J,1));
        ATOMIZE(LAM210(J,1));
        ATOMIZE(LAM211H(J,1));
        LET(PT(4,J)=LAM256(J,1));
        ATOMIZE(LAM256(J,1));
        LET(PT(5,J)=LAM257F(J,1));
        ATOMIZE(LAM257F(J,1));
        LET(PT(6,J)=LAM2578(J,1));
        ATOMIZE(LAM2578(J,1));
        LET(PT(7,J)=LAM2579E(J,1));
        ATOMIZE(LAM2579E(J,1));
        LET(PT(8,J)=LAM2579R(J,1));
        ATOMIZE(LAM2579R(J,1));
END LOOP16;
      */
/*                                                                 */
        K=12;  /*      PARTIALS W/R TO Q(J)                        */
        /*
LOOP17: DO I=PTNO TO PTNO; LET(I="I");
LOOP18: DO L=1 TO 3; LET(L="L");
LOOP19: DO J=1 TO 11; LET(J="J");
        LET(GG=REPLACE(DERIV(PT(I,L),Q(J),1),CHAN1));
        IF IDENT(GG,0) THEN GO TO AA;
        CHAREX(LINE=GG);
        CALL PUNCH1(LINE);
AA:     END LOOP19;
END LOOP18; END LOOP17;
        */
```

```
/*                                                                          */
/*                  SECOND PARTIALS W/R TO Q(J) AND Q(K)                    */
/*                                                                          */
          /*
LOOP20: DO I=PTNO TO PTNO; LET(I="I");
LOOP21: DO L=1 TO 3; LET(L="L");
LOOP22: DO J=1 TO 11; LET(J="J");
LOOP23: DO K=J TO 11; LET(K="K");
          LET(GG=REPLACE(DERIV(PT(I,L),Q(J),1,Q(K),1),CHAN1));
          IF IDENT(GG;0) THEN GO TO BB;
          CHAREX(LINE=GG);
          CALL PUNCH1(LINE);
BB: END LOOP23;
END LOOP22; END LOOP21; END LOOP20;
          */
/*                                                                          */
/*                  ANGULAR VELOCITIES                                      */
/*                                                                          */
/*       OBTAIN LAMB, LAM2, LAM5, LAM7, LAM9, AND LAM11 INVERSES            */
/*                                                                          */
LOOP50: DO I=1 TO 3; LET(I="I");
LOOP51: DO J=1 TO 3; LET(J="J");
          LET(LAMBI(I,J)=LAMB(J,I);
              LAM2I(I,J)=LAM2(J,I);
              LAM5I(I,J)=LAM5(J,I);
              LAM7I(I,J)=LAM7(J,I);
              LAM9I(I,J)=LAM9(J,I);
              LAM11I(I,J)=LAM11(J,I));
END LOOP51; END LOOP50;
          LET(VEC1(1,1)=0; VEC1(2,1)=QD(9); VEC1(3,1)=QD(10));
/*                                                                          */
/*                  OBTAIN WP VECTOR                                        */
/*                                                                          */
          /*
LOOP52: DO I=1 TO 3; LET(I="I");
LOOP53: DO J=1 TO 1; LET(J="J");
          LET(WP(I,J)=0);
```

```
LOOP54: DO K=1 TO 3; LET(K="K");
        LET(WP(I,J)=WP(I,J) + LAMBI(I,K)*VEC1(K,J));
END LOOP54; END LOOP53; END LOOP52;
        */
/*                                                                          */
/*          OBTAIN WQ VECTOR WHICH IS OMEGA(1,L)  L=1 TO 3                   */
/*                                                                          */
        /*
LOOP55: DO I=1 TO 3; LET(I="I");
LOOP56: DO J=1 TO 1; LET(J="J");
        LET(OMEGA(1,I)=0);
LOOP57: DO K=1 TO 3; LET(K="K");
        LET(OMEGA(1,I)=OMEGA(1,I) + LAM5I(I,K)*WP(K,J));
END LOOP57; END LOOP56; END LOOP55;
        LET(OMEGA(1,1)=OMEGA(1,1) + QD(6));
        */
/*                                                                          */
/*          OBTAIN WR VECTOR WHICH IS OMEGA(2,L)  L=1 TO 3                   */
/*                                                                          */
        /*
        LET(J=1);
LOOP58: DO I=1 TO 3; LET(I="I");
        LET(OMEGA(2,I)=0);
LOOP59: DO K=1 TO 3; LET(K="K");
        LET(OMEGA(2,I)=OMEGA(2,I) + LAM11I(I,K)*WP(K,J));
END LOOP59; END LOOP58;
        LET(OMEGA(2,1)=OMEGA(2,1) + QD(8));
        */
/*                                                                          */
/*          OBTAIN WS VECTOR WHICH IS OMEGA(3,L)  L=1 TO 3                   */
/*                                                                          */
        /*
LOOP60: DO I=1 TO 3; LET(I="I");
        LET(OMEGA(3,I)=0);
LOOP61: DO K=1 TO 3; LET(K="K");
        LET(OMEGA(3,I)=OMEGA(3,I) + LAM7I(I,K)*OMEGA(1,K));
END LOOP61; END LOOP60;
```

```
        LET(OMEGA(3,3)=OMEGA(3,3) + QD(11));
        */
/*                                                                    */
/*          OBTAIN WT VECTOR WHICH IS OMEGA(4,L)  L=1 TO 3           */
/*                                                                    */
        /*
LOOP62: DO I=1 TO 3; LET(I="I");
        LET(OMEGA(4,I)=0);
LOOP63: DO K=1 TO 3; LET(K="K");
        LET(OMEGA(4,I)=OMEGA(4,I) + LAM9I(I,K)*OMEGA(3,K));
END LOOP63; END LOOP62;
        LET(OMEGA(4,1)=OMEGA(4,1) + QD(7));
        */
/*                                                                    */
        K=12;  /*     PARTIALS W/R TO Q(J)                            */
/*                                                                    */
        /*
LOOP64: DO I=1 TO 4; LET(I="I");
LOOP65: DO L=1 TO 3; LET(L="L");
LOOP66: DO J=1 TO 11; LET(J="J");
        LET(GG=REPLACE(DERIV(OMEGA(I,L),Q(J),1),CHAN1));
        IF IDENT(GG;0) THEN GO TO CC;
        CHAREX(LINE=GG);
        CALL PUNCH1(LINE);
CC: END LOOP66;
END LOOP65; END LOOP64;
        */
/*                                                                    */
/*        FOR OMEGA PRINT OUT ONLY---NO DERIVATIVES                   */
/*                                                                    */
        J=12; K=12;  /*    NO DERIVATIVES                             */
        /*
LOOP67: DO I=1 TO 4; LET(I="I");
LOOP68: DO L=1 TO 3; LET(L="L");
        LET(GG=REPLACE(OMEGA(I,L),CHAN1));
        IF IDENT(GG,0) THEN GO TO DD;
        CHAREX(LINE=GG);
```

```
        CALL PUNCH1(LINE);
DD: END LOOP68;
END LOOP67;
        */
/*                                                                    */
/*                  PARTIALS W/R TO QD(K)                             */
/*                                                                    */
        /*
          J=12;
LOOP69: DO I=1 TO 4; LET(I="I");
LOOP70: DO K=1 TO 11; LET(K="K");
LOOP71: DO L=1 TO 3; LET(L="L");
        LET(GG=REPLACE(DERIU(OMEGA(I,L),QD(K),1),CHAN1));
        IF IDENT(GG,0) THEN GO TO EE;
        CHAREX(LINE=GG);
        CALL PUNCH1(LINE);
EE: END LOOP71;
END LOOP70; END LOOP69;
        */
/*                                                                    */
/*                  PARTIALS W/R TO Q(J) AND QD(K)                    */
/*                                                                    */
        /*
LOOP72: DO I=1 TO 4; LET(I="I");
LOOP73: DO J=1 TO 11; LET(J="J");
LOOP74: DO K=1 TO 11; LET(K="K");
LOOP75: DO L=1 TO 3; LET(L="L");
        LET(GG=REPLACE(DERIU(OMEGA(I,L),Q(J),1,QD(K),1),CHAN1));
        IF IDENT(GG,0) THEN GO TO FF;
        CHAREX(LINE=GG);
        CALL PUNCH1(LINE);
FF: END LOOP75;
END LOOP74; END LOOP73; END LOOP72;
        */
/*                                                                    */
/*                  POTENTIAL ENERGY                                  */
/*                                                                    */
```

```
/*                  DETERMINATION OF U2                                        */
/*                                                                             */
     /*
      I=2; K=12; L=0;
      I=1; K=12; L=0;
     */
     /*
      I=1; K=1; L=0;
LOOP76: DO III=1 TO 2; LET(III="III");
LOOP77: DO JJJ=1 TO 2; LET(JJJ="JJJ");
 L=L + 1;
 LET(STORE(1,1)=XL(III,JJJ); STORE(2,1)=XM(III,JJJ); STORE(3,1)=XM(III,
     JJJ));
 LET(TEMP=0);
LOOP78: DO KKK=1 TO 3; LET(KKK="KKK");
 LET(TEMP=TEMP + LAM25(1,KKK)*STORE(KKK,1));
END LOOP78;
     */
     /*
 LET(TEMP=TEMP + PT(1,3) + PT(2,3) - LAM3(3,1) - STORE(3,1));
     */
 LET(TEMP=TEMP + PT(1,1) + PT(2,1) - LAM3(1,1) - STORE(1,1));
     /*
 LET(TEMP=TEMP + PT(1,3) + PT(2,3));
     */
     /*
 LET(TEMP=TEMP*TEMP);
 LET(TEMP=TEMP*XK(III,JJJ)*0.5);
     */
     /*
LOOP79: DO J=1 TO 1; LET(J="J");
 LET(GG=REPLACE(TEMP,CHAN1));
 IF IDENT(GG;0) THEN GO TO PP.
 LET(GG=REPLACE(GG,CHAN3)).
 LET(GG=REPLACE(GG,CHAN28));
 LET(GG=REPLACE(GG,CHAN29));
     */
```

```
      /*
LOOP79  DO J=1 TO 11; LET(J='J');
 LET(GG=REPLACE(DERIV(TEMP,Q(J),1),CHAN1));
 IF IDENT(GG;0) THEN GO TO PP;
 LET(GG=REPLACE(GG,CHAN3));
      */
      /*
 LET(GG=REPLACE(GG,CHAN14));
 LET(GG=REPLACE(GG,CHAN15));
 LET(GG=REPLACE(GG,CHAN16));
 LET(GG=REPLACE(GG,CHAN17));
      */
      /*
 CHAREX(LINE=GG);
 CALL PUNCH1(LINE);
PP: END LOOP79;
END LOOP77; END LOOP76;
      */
/*                                                                    */
/*                   DETERMINATION OF U3                              */
/*                                                                    */
      /*
 LET(L1FIRST(1,1)=A1 - G1 - D1P;
     L1FIRST(2,1)=A2 - G2 - D2P;
     L1FIRST(3,1)=A3 - G3 - D3P);
 LET(L1SEC(1,1)=A1BAR - G1 + D1P;
     L1SEC(2,1)=A2 - G2 - D2P;
     L1SEC(3,1)=A3 - G3 - D3P);
 LET(L1F=L1FIRST(1,1)**2 + L1FIRST(2,1)**2 + L1FIRST(3,1)**2);
 LET(L1F=L1F**0.5);
 LET(L1S=L1SEC(1,1)**2 + L1SEC(2,1)**2 + L1SEC(3,1)**2);
 LET(L1S=L1S**0.5);
      */
      /*
 LET(SAVEF(1,1)=A1; SAVEF(2,1)=A2; SAVEF(3,1)=A3);
 LET(SAVES(1,1)=A1BAR; SAVES(2,1)=A2; SAVES(3,1)=A3);
      */
```

```
      /*
 LET(SAVEF(1,1)=B1; SAVEF(2,1)=B2; SAVEF(3,1)=B3);
 LET(SAVES(1,1)=B1BAR; SAVES(2,1)=B2; SAVES(3,1)=B3);
      */
      /*
 LET(J=1);
LOOP80: DO I=1 TO 3; LET(I="I");
 LET(L2FIRST(I,J)=0; L2SEC(I,J)=0);
LOOP81: DO K=1 TO 3; LET(K="K");
 LET(L2FIRST(I,J)=L2FIRST(I,J) + LAM5(I,K)*SAVEF(K,J));
 LET(L2SEC(I,J)=L2SEC(I,J) + LAM5(I,K)*SAVES(K,J));
END LOOP81; END LOOP80;
LOOP82: DO I=1 TO 3; LET(I="I");
 LET(L2FIRST(I,J)=L2FIRST(I,J) + LAM4(I,J) - LAM10(I,J));
 LET(L2SEC(I,J)=L2SEC(I,J) + LAM4(I,J) - LAM10(I,J));
END LOOP82;
LOOP83: DO I=1 TO 3; LET(I="I");
 LET(L2FT(I,J)=0; L2SC(I,J)=0);
LOOP84: DO K=1 TO 3; LET(K="K");
 LET(L2FT(I,J)=L2FT(I,J) + LAM11I(I,K)*L2FIRST(K,J));
 LET(L2SC(I,J)=L2SC(I,J) + LAM11I(I,K)*L2SEC(K,J));
END LOOP84; END LOOP83;
 LET(L2FT(1,1)=L2FT(1,1) - D1P;
     L2FT(2,1)=L2FT(2,1) - D2P;
     L2FT(3,1)=L2FT(3,1) - D3P);
 LET(L2SC(1,1)=L2SC(1,1) + D1P;
     L2SC(2,1)=L2SC(2,1) - D2P;
     L2SC(3,1)=L2SC(3,1) - D3P);
      */
      /*
 LET(L2FT(1,1)=L2FT(1,1) - C1P;
     L2FT(2,1)=L2FT(2,1) - C2P;
     L2FT(3,1)=L2FT(3,1) - C3P);
 LET(L2SC(1,1)=L2SC(1,1) + C1P;
     L2SC(2,1)=L2SC(2,1) - C2P;
     L2SC(3,1)=L2SC(3,1) - C3P);
      */
```

```
     /*
LET(L2F=L2FT(1,1)**2 + L2FT(2,1)**2 + L2FT(3,1)**2);
LET(L2S=L2SC(1,1)**2 + L2SC(2,1)**2 + L2SC(3,1)**2);
LET(L2F=L2F**0.5; L2S=L2S**0.5);
     */
     /*
LET(U3=0.5*XKK1*(L1F - L2F)**2); L=1;
     */
     /*
LET(U3=0.5*XKK2*(L1S - L2S)**2); L=2;
     */
     /*
LET(U3=L2F); L=1;
     */
     /*
LET(U3=L2S); L=2;
     */
     /*
I=2; K=12;
LOOP85: DO J=1 TO 11; LET(J="J");
LET(GG=REPLACE(DERIV(U3,Q(J),1),CHAN1));
IF IDENT(GG;0) THEN GO TO QQ;
     */
     /*
LET(GG=REPLACE(GG,CHAN7));
LET(GG=REPLACE(GG,CHAN8));
LET(GG=REPLACE(GG,CHAN9));
LET(GG=REPLACE(GG,CHAN10));
LET(GG=REPLACE(GG,CHAN11));
LET(GG=REPLACE(GG,CHAN12));
LET(GG=REPLACE(GG,CHAN13));
     */
     /*
LET(GG=REPLACE(GG,CHAN19));
LET(GG=REPLACE(GG,CHAN20));
LET(GG=REPLACE(GG,CHAN21));
LET(GG=REPLACE(GG,CHAN22));
```

```
 LET(GG=REPLACE(GG,CHAN23));
     */
     /*
 LET(GG=REPLACE(GG,CHAN24));
 LET(GG=REPLACE(GG,CHAN25));
 LET(GG=REPLACE(GG,CHAN26));
 LET(GG=REPLACE(GG,CHAN27));
 CHAREX(LINE=GG);
 CALL PUNCH1(LINE);
QQ: END LOOP85;
     */
/*                                                                     */
/*                  DETERMINATION OF U4                                */
/*                                                                     */
    /*
 LET(STORE(1,1)=ALPHA1; STORE(2,1)=ALPHA2; STORE(3,1)=ALPHA3);
 LET(STORE(1,1)=-ALPHA1);
LOOP86: DO I=1 TO 3; LET(I="I");
 LET(DELTA(I,1)=STORE(I,1) - LAM1(I,1));
END LOOP86;
LOOP87: DO I=1 TO 3; LET(I="I");
 LET(DELTA1(I,1)=0);
LOOP88: DO K=1 TO 3; LET(K="K");
 LET(DELTA1(I,1)=DELTA1(I,1) + LAM2I(I,K)*DELTA(K,1));
END LOOP88; END LOOP87;
LOOP89: DO I=1 TO 3; LET(I="I");
 LET(DELTA1(I,1)=DELTA1(I,1) - LAM3(I,1) - LAM10(I,1));
END LOOP89;
LOOP90: DO I=1 TO 3; LET(I="I");
 LET(DELTA(I,1)=0);
LOOP91: DO K=1 TO 3; LET(K="K");
 LET(DELTA(I,1)=DELTA(I,1) + LAM11I(I,K)*DELTA1(K,1));
END LOOP91; END LOOP90;
LOOP92: DO I=1 TO 3; LET(I="I");
 LET(DELTA(I,1)=DELTA(I,1) - STORE(I,1) + LAM1(I,1) + LAM3(I,1) +
     LAM10(I,1));
END LOOP92;
```

```
      */
      /*
 LET(DELTA(2,1)=DELTA(2,1)*DELTA(2,1)*XKY1/2); L=1;
      */
      /*
 LET(DELTA(2,1)=DELTA(2,1)*DELTA(2,1)*XKY2/2); L=2;
 I=4; K=12;
LOOP93: DO J=1 TO 11; LET(J="J");
 LET(GG=REPLACE(DERIV(DELTA(2,1),Q(J),1),CHAN1));
 IF IDENT(GG;0) THEN GO TO RRR;
 LET(GG=REPLACE(GG,CHAN18));
 CHAREX(LINE=GG);
 CALL PUNCH1(LINE);
RRR: END LOOP93;
      */
/*                                                                       */
/*                    GENERALIZED FORCES                                 */
/*                                                                       */
      /*
      LET(SAVE(1,1)=XIB; SAVE(2,1)=Q(1) + EB; SAVE(3,1)=ZETAB);
      LET(SAVE(1,1)=XIR; SAVE(2,1)=Q(1) + ER; SAVE(3,1)=ZETAR);
      LET(SAVE(1,1)=XIC; SAVE(2,1)=Q(1) + EC; SAVE(3,1)=ZETAC);
      LET(SAVE(1,1)=0; SAVE(2,1)=-BOFT; SAVE(3,1)=0);
      LET(SAVE(1,1)=0; SAVE(2,1)=ROFT; SAVE(3,1)=0);
      */
      LET(SAVE(1,1)=0; SAVE(2,1)=COFT; SAVE(3,1)=0);
      J=1; LET(J="J");
LOOP94: DO I=1 TO 3; LET(I="I");
      LET(LAM2579S(I,J)=0);
LOOP95: DO K=1 TO 3; LET(K="K");
      LET(LAM2579S(I,J)=LAM2579S(I,J) + LAM2579(I,K)*SAVE(K,J));
END LOOP95; END LOOP94;
/*                                                                       */
/*                    EVALUATE PG(I,J)'S                                 */
/*                                                                       */
      K=12; /*  PARTIALS W/R TO Q(J)                                     */
LOOP96: DO I=PTNO TO PTNO; LET(I="I");
```

```
LOOP97: DO L=1 TO 3; LET(L="L");
LOOP98: DO J=12 TO 12; LET(J="J");
     /*
     LET(GG=REPLACE(DERIV(LAM2579S(L,1),Q(J),1),CHAN1));
     */
     LET(GG=REPLACE(LAM2579S(L,1),CHAN1));
     IF IDENT(GG;0) THEN GO TO UU;
     LET(GG=REPLACE(GG,CHAN3));
     LET(GG=REPLACE(GG,CHAN30));
     CHAREX(LINE=GG);
     CALL PUNCH1(LINE);
UU: END LOOP98;
    END LOOP97; END LOOP96;
/*                                                                 */
      XYZ: PUT FILE(CARD) EDIT(CARDS) (SKIP(3),X(6),F(5));
/*                                                                 */
 PUNCH1:   PROCEDURE(LINE);
           DCL LINE CHAR(600) VAR, A CHAR(66) VAR;
           JJ = 1;
           NX = LENGTH(LINE);
           NNX = NX;
           IF NNX <= 53 THEN GO TO ONE;
           A = SUBSTR(LINE,3,51);
           PUT FILE(CARD) EDIT('P','G','(',I,',',J,',',K,',',L,')',A)
                 (SKIP(1),X(6),A(1),A(1),A(1),4(F(2),A(1)),A);
           CARDS = CARDS+1;
           NNX = NX - 53;
 THREE:    LL = 66*(JJ-1)+54;
           IF NNX <= 66 THEN GO TO TWO;
           A = SUBSTR(LINE,LL,66);
           PUT FILE(CARD) EDIT('1',A)(SKIP(1),X(5),A(1),A);
           CARDS = CARDS + 1;
           JJ = JJ + 1;
           NNX = NNX-66;
           GO TO THREE;
   ONE:    A = SUBSTR(LINE,3,NNX-2);
           PUT FILE(CARD) EDIT('P','G','(',I,',',J,',',K,',',L,')',A)
```

```
                   (SKIP(1),X(6),A(1),A(1),A(1),4(F(2),A(1)),A);
             CARDS = CARDS+1;
              GO TO TERM;
     TWO:     A = SUBSTR(LINE,LL,NNX);
              PUT FILE(CARD) EDIT('1',A)(SKIP(1),X(5),A(1),A);
              CARDS = CARDS + 1;
     TERM:    END PUNCH1;
    END START;
END OF DATA
```

APPENDIX D

THE SQUARE-ROOT METHOD

APPENDIX D

The Square-Root Method

This appendix presents the method that was used to decouple the acceleration terms, i.e., solve the matrix equation $A\ddot{X} = B$. The method can also be used to solve any system of linear equations of the form $AX = B$ if the A matrix is symmetric, which is the case when solving Lagrange's equations of motion.

The algorithm used is called the square-root method. In case the **coefficient** matrix of a system is symmetric, finding the solution is made more expedient by taking advantage of the symmetry. Using the equations presented in Reference 1 produces pure imaginary numbers in the computational scheme and so a modification to that algorithm is given here to alleviate the problem of imaginary numbers.

A FORMAC program was developed using the modified square-root method to solve a system of equations either numerically or symbolically. The program is listed here along with results, both numeric and symbolic. Cards can be automatically punched in FORTRAN format.

FORMAC can be used to obtain symbolic solutions in problem areas which heretofore could only be approached numerically. This is well exemplified in this appendix.

The solution of a system using the square-root method reduces to the solution of two triangular systems. For the equation

$$AX = F$$

the algorithm according to Reference 1 is

$$s_{11} = \sqrt{a_{11}}\ , \qquad s_{1j} = a_{1j}/s_{11}$$

$$s_{11} = \left[a_{ii} - \sum_{m=1}^{i-1} s_{mi}^2\right]^{\frac{1}{2}},\ i > 1;\ s_{ij} = \left[a_{ij} - \sum_{m=1}^{i-1} s_{mi}\, s_{mj}\right]/s_{ii},\ j > i$$

$$s_{ij} = 0,\quad i > j$$

$$k_1 = \frac{f_1}{s_{11}}, \qquad k_i = \frac{f_i - \sum_{m=1}^{i-1} s_{mi} k_m}{s_{ii}} \; ; \; i > 1$$

The final solution is found by the formulas

$$x_n = \frac{k_n}{s_{nn}}, \quad x_i = \frac{k_i - \sum_{m=i+1}^{n} s_{im} x_m}{s_{ii}} \; ; \; i < n$$

In case the elements of the matrix are such that radicands of the expression s_{ii} are negative, pure imaginary numbers appear in the r w for which $s_{ii}^2 < 0$. To alleviate this problem, the following definitions are made

$$s_{11} = a_{11} \qquad , \qquad s_{1j} = a_{ij}$$

$$c_1 = 1/s_{11} \qquad , \qquad c_i = 1/s_{ii}; \; i > 1$$

$$s_{ii} = a_{ii} - \sum_{m=1}^{i-1} s_{mi}^2 \, c_m, \; i > 1$$

$$s_{ij} = a_{ij} - \sum_{m=1}^{i-1} s_{mi} \, s_{mj} \, c_m, \; j > i$$

$$s_{ij} = 0, \quad i > j$$

$$k_1 = f_1 \qquad , \qquad k_i = f_i - \sum_{m=1}^{i-1} s_{mi} \, k_m \, c_m$$

$$x_n = k_n c_n \qquad , \qquad x_i = c_i k_i - \sum_{m=i+1}^{n} s_{im} \, x_m \, c_i; \; i < n$$

With this method, approximately $n^2/2$ elements of the S matrix and 2n components of the vectors K and X are recorded. The square-root method is widely employed where the solution of symmetric systems is called for and it is recommended as one of the most efficient methods.

A listing of the FORMAC program is given below. To solve the system of equations AX = F, only the value N representing the size of the square matrix A and the variable SOLVE must be changed to produce either a numeric or symbolic solution.

For N = 11 and SOLVE = NO, the output is contained in the FORTRAN listing under the subroutine named SOLVE in Appendix F.

```
START  PROCEDURE OPTIONS(MAIN);
       DCL LINE CHAR(2000) VAR;
       FORMAC_OPTIONS;
       OPTSET(NOEDIT);
       OPTSET(LINELENGTH=80);
/*                                                                    */
/*                N IS THE ORDER OF THE COEFFICIENTS MATRIX           */
/*                                                                    */
     N=11; LET(N="N");
     NMINUS1=N - 1;
/*                                                                    */
/*      SET SOLVE=YES FOR NUMERIC SOLUTION. OTHERWISE SOLVE=NO        */
/*                                                                    */
     LET(SOLVE=NO);
     IF IDENT(SOLVE,NO) THEN GO TO HH;
/*                                                                    */
/*           DETERMINATION OF S(I,J)  EQUATION 3                      */
/*                                                                    */
HH: LET(S(1,1)=A(1,1));
LOOP10: DO J=2 TO N; LET(J="J");
     LET(S(1,J)=A(1,J));
END LOOP10;
LOOP40: DO I=1 TO N; LET(I="I");
     CHAREX(LINE=S(1,I));
     CALL PUNCH1(LINE);
     IF IDENT(SOLVE,YES) THEN GO TO AA;
     ATOMIZE(S(1,I));
AA: END LOOP40;
     LET(C(1)=1./S(1,1));
     CHAREX(LINE=C(1));
     CALL PUNCH1(LINE);
     IF IDENT(SOLVE,YES) THEN GO TO LOOP11;
     ATOMIZE(C(1));
LOOP11: DO I=2 TO NMINUS1; LET(I="I");
     IMINUS1=I - 1;
     IPLUS1=I + 1;
     LET(SUM1=0);
```

```
LOOP13  DO J=IPLUS1 TO N, LET(J='J'),
     LET(SUM2=0)
LOOP14  DO L=1 TO IMINUS1, LET(L='L');
     LET(SUM2=SUM2 + S(L,I)*S(L,J)*C(L)),
END LOOP14,
     LET(S(I,J)=A(I,J) - SUM2);
     CHAREX(LINE=S(I,J));
     CALL PUNCH1(LINE);
     IF IDENT(SOLVE,YES) THEN GO TO BB;
     ATOMIZE(S(I,J));
BB: END LOOP13;
LOOP12: DO L=1 TO IMINUS1, LET(L='L');
     LET(SUM1=SUM1 + C(L)*S(L,I)**2);
END LOOP12;
     LET(S(I,I)=A(I,I) - SUM1);
     LET(C(I)=1./S(I,I));
     CHAREX(LINE=C(I));
     CALL PUNCH1(LINE);
     IF IDENT(SOLVE,YES) THEN GO TO CC;
     ATOMIZE(C(I));
CC: ATOMIZE(S(I,I)),
END LOOP11;
     LET(SUM1=0);
LOOP19: DO L=1 TO NMINUS1; LET(L='L');
     LET(SUM1=SUM1 + C(L)*S(L,N)**2);
END LOOP19;
     LET(S(N,N)=A(N,N) - SUM1);
     LET(C(N)=1./S(N,N));
     CHAREX(LINE=C(N));
     CALL PUNCH1(LINE);
     IF IDENT(SOLVE;YES) THEN GO TO DD;
     ATOMIZE(C(N)),
DD: ATOMIZE(S(N,N)),
/*                                                        */
/*          DETERMINATION OF THE K(I)   EQUATION 4        */
/*                                                        */
     LET(XK(1)=F(1)),
```

```
        CHAREX(LINE=XK(1));
        CALL PUNCH1(LINE);
        IF IDENT(SOLVE;YES) THEN GO TO LOOP5;
        ATOMIZE(XK(1));
 LOOP5: DO I=2 TO N; LET(I="I");
        LET(SUM=0);
        IMINUS1=I - 1;
 LOOP4: DO L=1 TO IMINUS1; LET(L="L");
        LET(SUM=SUM + S(L,I)*XK(L)*C(L));
 END LOOP4;
        LET(XK(I)=F(I) - SUM);
        CHAREX(LINE=XK(I));
        CALL PUNCH1(LINE);
        IF IDENT(SOLVE;YES) THEN GO TO FF;
        ATOMIZE(XK(I));
 FF: END LOOP5;
 /*                                                              */
 /*          SOLUTION FORMULAS    EQUATION 5                     */
 /*                                                              */
        LET(X(N)=XK(N)*C(N));
        CHAREX(LINE=X(N));
        CALL PUNCH1(LINE);
        IF IDENT(SOLVE;YES) THEN GO TO LOOP70;
        ATOMIZE(X(N));
 LOOP70: DO M=1 TO NMINUS1; LET(M="M");
        I=N - M; LET(I="I");
        IPLUS1=I + 1;
        LET(SUM=0);
 LOOP71: DO L=IPLUS1 TO N; LET(L="L");
        LET(SUM=SUM + C(I)*S(I,L)*X(L));
 END LOOP71;
        LET(X(I)=XK(I)*C(I) - SUM);
        CHAREX(LINE=X(I));
        CALL PUNCH1(LINE);
        IF IDENT(SOLVE;YES) THEN GO TO GG;
        ATOMIZE(X(I));
 GG: END LOOP70;
```

```
/*
PUNCH1: PROCEDURE(LINE);
        DCL LINE CHAR(2000) VAR, A CHAR(66) VAR;
        JJ=1;
        NX=LENGTH(LINE);
        NNX=NX;
        IF NNX <= 66 THEN GO TO ONE;
        A=SUBSTR(LINE,1,66);
        PUT FILE(CARD) EDIT(A) (SKIP(1),X(6),A);
        NNX=NX - 66;
THREE:  LL=66*(JJ-1) + 67;
        IF NNX <= 66 THEN GO TO TWO;
        A=SUBSTR(LINE,LL,66);
        PUT FILE(CARD) EDIT('1',A)(SKIP(1),X(5),A(1),A);
        JJ=JJ + 1;
        NNX=NNX - 66;
        GO TO THREE;
ONE:    A=SUBSTR(LINE,1,NNX);
        PUT FILE(CARD) EDIT(A) (SKIP(1),X(6),A);
        GO TO TERM;
TWO:    A=SUBSTR(LINE,LL,NNX);
       PUT FILE(CARD) EDIT('1',A)(SKIP(1),X(5),A(1),A);
TERM:   END PUNCH1;
END START;
END OF DATA
```

For N = 3 and SOLVE = NO the following calculations as computed by FORMAC in symbolic form solve the system of equations with symmetric matrix A.

$$A_{3x3}\ X_{3x1} = F_{3x1}$$

```
POTENT CARDS DATA
      S(1,1) = A(1,1)
      S(1,2) = A(1,2)
      S(1,3) = A(1,3)
      C(1) = S(1,1)**(-1 )
      S(2,3) = A(2,3)-S(1,2)*S(1,3)*C(1)
      C(2) = (A(2,2)-S(1,2)**2*C(1))**(-1 )
      C(3) = (A(3,3)-S(2,3)**2*C(2)-S(1,3)**2*C(1))**(-1 )
      XK(1) = F(1)
      XK(2) = F(2)-XK(1)*S(1,2)*C(1)
      XK(3) = F(3)-XK(1)*S(1,3)*C(1)-XK(2)*S(2,3)*C(2)
      X(3) = XK(3)*C(3)
      X(2) = XK(2)*C(2)-X(3)*S(2,3)*C(2)
      X(1) = XK(1)*C(1)-X(3)*S(1,3)*C(1)-X(2)*S(1,2)*C(1)
READY
```

For N = 15 and SOLVE = YES (with the appropriate A matrix now defined), the following numeric solution is given. The solution of course is the vector x(1) through x(15).

```
START: PROCEDURE OPTIONS(MAIN);
       DCL LINE CHAR(900) VAR;
       FORMAC_OPTIONS;
       OPTSET(NOEDIT);
       OPTSET(LINELENGTH=80);
/*                                                                      */
/*                   N IS THE ORDER OF THE COEFFICIENTS MATRIX          */
/*                                                                      */
     N=15; LET(N="N");
     NMINUS1=N - 1;
/*                                                                      */
/*       SET SOLVE=YES FOR NUMERIC SOLUTION, OTHERWISE SOLVE=NO         */
/*                                                                      */
     LET(SOLVE=YES);
     IF IDENT(SOLVE;NO) THEN GO TO HH;
/*                                                                      */
LOOP90: DO I=1 TO 15; LET(I="I");
LOOP91: DO J=1 TO 15; LET(J="J");
     LET(F(I)=0);
     LET(A(I,J)=0.);
END LOOP91; END LOOP90;
     LET(A(1,1)=18; A(1,2)=-8; A(1,3)=1; A(1,6)=-8; A(1,7)=2;A(1,11)=1;
A(2,2)=19;A(2,3)=-8;A(2,4)=1;A(2,6)=2;A(2,7)=-8;A(2,8)=2;A(2,12)=1;
A(3,3)=19;A(3,4)=-8;A(3,5)=1;A(3,7)=2;A(3,8)=-8;A(3,9)=2;A(3,13)=1);
     LET(A(4,4)=19;A(4,5)=-8;A(4,8)=2;A(4,9)=-8;A(4,10)=2;A(4,14)=1;
A(5,5)=18;A(5,9)=2;A(5,10)=-8;A(5,15)=1;
A(6,6)=19;A(6,7)=-8;A(6,8)=1;A(6,11)=-8;A(6,12)=2);
     LET(A(7,7)=20;A(7,8)=-8;A(7,9)=1;A(7,11)=2;A(7,12)=-8;A(7,13)=2;
A(8,8)=20;A(8,9)=-8;A(8,10)=1;A(8,12)=2;A(8,13)=-8;A(8,14)=2;
A(9,9)=20;A(9,10)=-8;A(9,13)=2;A(9,14)=-8;A(9,15)=2);
     LET(A(10,10)=19;A(10,14)=2;A(10,15)=-8;
A(11,11)=18;A(11,12)=-8;A(11,13)=1;
A(12,12)=19;A(12,13)=-8;A(12,14)=1);
     LET(A(13,13)=19;A(13,14)=-8;A(13,15)=1;
A(14,14)=19;A(14,15)=-8;
A(15,15)=18);
LOOP60: DO I=1 TO 15; LET(I="I");
```

```
LOOP61: DO J=1 TO 15; LET(J="J");
        LET(A(J,I)=A(I,J));
END LOOP61;END LOOP60;
        LET(F(8)=-.25; F(9)=-.5; F(10)=-.5;F(13)=-.5;
            F(14)=-1; F(15)=-1);
/*                                                                  */
/*              DETERMINATION OF S(I,J)  EQUATION 3                 */
/*                                                                  */
HH: LET(S(1,1)=A(1,1));
LOOP10: DO J=2 TO N; LET(J="J");
        LET(S(1,J)=A(1,J));
END LOOP10;
LOOP40: DO I=1 TO N; LET(I="I");
        CHAREX(LINE=S(1,I));
        CALL PUNCH1(LINE);
        IF IDENT(SOLVE,YES) THEN GO TO AA;
        ATOMIZE(S(1,I));
AA: END LOOP40;
        LET(C(1)=1./S(1,1));
        CHAREX(LINE=C(1));
        CALL PUNCH1(LINE);
        IF IDENT(SOLVE,YES) THEN GO TO LOOP11;
        ATOMIZE(C(1));
LOOP11: DO I=2 TO NMINUS1; LET(I="I");
        IMINUS1=I - 1;
        IPLUS1=I + 1;
        LET(SUM1=0);
LOOP13: DO J=IPLUS1 TO N; LET(J="J");
        LET(SUM2=0);
LOOP14: DO L=1 TO IMINUS1; LET(L="L");
        LET(SUM2=SUM2 + S(L,I)*S(L,J)*C(L));
END LOOP14;
        LET(S(I,J)=A(I,J) - SUM2);
        CHAREX(LINE=S(I,J));
        CALL PUNCH1(LINE);
        IF IDENT(SOLVE,YES) THEN GO TO BB;
        ATOMIZE(S(I,J));
```

POTENT.CARDS.DATA

```
S(1,1) = 18
S(1,2) = -8
S(1,3) = 1
S(1,4) = 0
S(1,5) = 0
S(1,6) = -8
S(1,7) = 2
S(1,8) = 0
S(1,9) = 0
S(1,10) = 0
S(1,11) = 1
S(1,12) = 0
S(1,13) = 0
S(1,14) = 0
S(1,15) = 0
C(1) = .05555555
S(2,3) = -7.55555555
S(2,4) = 1
S(2,5) = 0
S(2,6) = -1.55555555
S(2,7) = -7.11111111
S(2,8) = 2
S(2,9) = 0
S(2,10) = 0
S(2,11) = .44444444
S(2,12) = 1
S(2,13) = 0
S(2,14) = 0
S(2,15) = 0
C(2) = .0647482
S(3,4) = -7.51079136
S(3,5) = 1
S(3,6) = -.31654676
S(3,7) = -1.58992805
S(3,8) = -7.02158273
```

```
S(3,9) = 2
S(3,10) = 0
S(3,11) = .1618705
S(3,12) = .48920863
S(3,13) = 1
S(3,14) = 0
S(3,15) = 0
C(3) = .0655815
S(4,5) = -7.50743099
S(4,6) = -.05520169
S(4,7) = -.32271762
S(4,8) = -1.5881104
S(4,9) = -7.01486199
S(4,10) = 2
S(4,11) = .05095541
S(4,12) = .1762208
S(4,13) = .492569
S(4,14) = 1
S(4,15) = 0
C(4) = .06563545
S(5,6) = -.00644122
S(5,7) = -.0547504
S(5,8) = -.32206119
S(5,9) = -1.58776167
S(5,10) = -7.01449275
S(5,11) = .01449275
S(5,12) = .0547504
S(5,13) = .17713365
S(5,14) = .49275362
S(5,15) = 1
C(5) = .07024886
S(6,7) = -7.86153846
S(6,8) = 1.04977375
S(6,9) = .01538461
S(6,10) = .00407239
S(6,11) = -7.50723981
S(6,12) = 2.11153846
```

```
S(6,13) = .02262443
S(6,14) = .00384615
S(6,15) = .45248868E-03
C(6) = .06544076
S(7,8) = -7.30608191
S(7,9) = 1.061761
S(7,10) = .01747972
S(7,11) = -1.75067422
S(7,12) = -6.39830259
S(7,13) = 2.12702409
S(7,14) = .02505563
S(7,15) = .00407894
C(7) = .08139158
S(8,9) = -7.21583074
S(8,10) = 1.05988836
S(8,11) = -.50268795
S(8,12) = -1.83444748
S(8,13) = -6.22087462
S(8,14) = 2.13001981
S(8,15) = .0250189
C(8) = .08390573
S(9,10) = -7.22134415
S(9,11) = -.14165743
S(9,12) = -.53678325
S(9,13) = -1.83487816
S(9,14) = -6.19716491
S(9,15) = 2.12633319
C(9) = .08424474
S(10,11) = -.03653049
S(10,12) = -.15103259
S(10,13) = -.54344795
S(10,14) = -1.84803156
S(10,15) = -6.21589396
C(10) = .09264806
S(11,12) = -7.99324607
S(11,13) = 1.01562838
S(11,14) = .01124346
```

```
S(11,15) = .00517853
C(11) = .07158643
S(12,13) = -7.40085196
S(12,14) = 1.02724051
S(12,15) = .01420665
C(12) = .09605265
S(13,14) = -7.25258412
S(13,15) = 1.02535098
C(13) = .10357286
S(14,15) = -7.22443817
C(14) = .10598317
C(15) = .12006693
XK(1) = 0
XK(2) = 0
XK(3) = 0
XK(4) = 0
XK(5) = 0
XK(6) = 0
XK(7) = 0
XK(8) = -.25
XK(9) = -.65136239
XK(10) = -.87403035
XK(11) = -.02127602
XK(12) = -.09234004
XK(13) = -.83928054
XK(14) = -2.06634548
XK(15) = -2.87901314
X(15) = -.34567428
X(14) = -.48366991
X(13) = -.41353535
X(12) = -.25464496
X(11) = -.11664933
X(10) = -.38763933
X(9) = -.55812172
X(8) = -.49721743
X(7) = -.32342785
X(6) = -.15294545
```

```
  X(5) = -.21855807
  X(4) = -.32456406
  X(3) = -.30166405
  X(2) = -.20665022
  X(1) = -.10064424
READY
```

APPENDIX E

REMOVAL OF IMBEDDED TERMS

APPENDIX E

Removal of Imbedded Terms

This appendix displays the reduction in the number of arithmetic operations that were accomplished by using FORMAC. CHAIN and REPLACE operations enabled character strings involving algebraic expressions to be replaced by new variable names and thus eliminate millions of arithmetic operations during the execution of the program.

The first five equations (53 lines of FORTRAN formated output) represent the partial derivatives of U_3 (part 2 of two parts) in the potential energy with only the sines and cosines replaced, i.e., S6 = SIN (Q(6)), C8 = COS(Q(8)), etc. The last five equations (7 lines) are the result of removing imbedded terms.

A similar type of reduction in the number of operations was performed on all expressions before the FORMAC output was punched on cards.

```
 PU( 3, 2,12, 2) = -XKK2*(-C8*(-D2P + C8*(-G2-Q(2) + Q(4) + A2*C6-A
13*S6) + S8*(-G3 + Q(5) + A2*S6 + A3*C6))*2 + S8*(-D3P + C8*(-G3 +
1Q(5) + A2*S6 + A3*C6)-S8*(-G2-Q(2) + Q(4) + A2*C6-A3*S6))*2)*((D1P
1 + A1BAR-G1)**2 + (-D2P + C8*(-G2-Q(2) + Q(4) + A2*C6-A3*S6) + S8*
1(-G3 + Q(5) + A2*S6 + A3*C6))**2 + (-D3P + C8*(-G3 + Q(5) + A2*S6
1+ A3*C6)-S8*(-G2-Q(2) + Q(4) + A2*C6-A3*S6))**2)**(-.5 )*(((D1P +
1A1BAR-G1)**2 + (-D2P-G2 + A2)**2 + (-D3P-G3 + A3)**2)**.5-((D1P +
1A1BAR-G1)**2 + (-D2P + C8*(-G2-Q(2) + Q(4) + A2*C6-A3*S6) + S8*(-G
13 + Q(5) + A2*S6 + A3*C6))**2 + (-D3P + C8*(-G3 + Q(5) + A2*S6 + A
13*C6)-S8*(-G2-Q(2) + Q(4) + A2*C6-A3*S6))**2)**.5)*.5
 PU( 3, 4,12, 2) = -XKK2*(C8*(-D2P + C8*(-G2-Q(2) + Q(4) + A2*C6-A3
1*S6) + S8*(-G3 + Q(5) + A2*S6 + A3*C6))*2-S8*(-D3P + C8*(-G3 + Q(5
1) + A2*S6 + A3*C6)-S8*(-G2-Q(2) + Q(4) + A2*C6-A3*S6))*2)*((D1P +
1A1BAR-G1)**2 + (-D2P + C8*(-G2-Q(2) + Q(4) + A2*C6-A3*S6) + S8*(-G
13 + Q(5) + A2*S6 + A3*C6))**2 + (-D3P + C8*(-G3 + Q(5) + A2*S6 + A
13*C6)-S8*(-G2-Q(2) + Q(4) + A2*C6-A3*S6))**2)**(-.5 )*(((D1P + A1B
1AR-G1)**2 + (-D2P-G2 + A2)**2 + (-D3P-G3 + A3)**2)**.5-((D1P + A1B
1AR-G1)**2 + (-D2P + C8*(-G2-Q(2) + Q(4) + A2*C6-A3*S6) + S8*(-G3 +
1 Q(5) + A2*S6 + A3*C6))**2 + (-D3P + C8*(-G3 + Q(5) + A2*S6 + A3*C
16)-S8*(-G2-Q(2) + Q(4) + A2*C6-A3*S6))**2)**.5)*.5
 PU( 3, 5,12, 2) = -XKK2*(C8*(-D3P + C8*(-G3 + Q(5) + A2*S6 + A3*C6
1)-S8*(-G2-Q(2) + Q(4) + A2*C6-A3*S6))*2 + S8*(-D2P + C8*(-G2-Q(2)
1+ Q(4) + A2*C6-A3*S6) + S8*(-G3 + Q(5) + A2*S6 + A3*C6))*2)*((D1P
1+ A1BAR-G1)**2 + (-D2P + C8*(-G2-Q(2) + Q(4) + A2*C6-A3*S6) + S8*(
1-G3 + Q(5) + A2*S6 + A3*C6))**2 + (-D3P + C8*(-G3 + Q(5) + A2*S6 +
1 A3*C6)-S8*(-G2-Q(2) + Q(4) + A2*C6-A3*S6))**2)**(-.5 )*(((D1P + A
11BAR-G1)**2 + (-D2P-G2 + A2)**2 + (-D3P-G3 + A3)**2)**.5-((D1P + A
11BAR-G1)**2 + (-D2P + C8*(-G2-Q(2) + Q(4) + A2*C6-A3*S6) + S8*(-G3
1 + Q(5) + A2*S6 + A3*C6))**2 + (-D3P + C8*(-G3 + Q(5) + A2*S6 + A3
1*C6)-S8*(-G2-Q(2) + Q(4) + A2*C6-A3*S6))**2)**.5)*.5
 PU( 3, 6,12, 2) = -XKK2*((-D2P + C8*(-G2-Q(2) + Q(4) + A2*C6-A3*S6
1) + S8*(-G3 + Q(5) + A2*S6 + A3*C6))*(C8*(-A2*S6-A3*C6) + S8*(A2*C
16-A3*S6))*2 + (-D3P + C8*(-G3 + Q(5) + A2*S6 + A3*C6)-S8*(-G2-Q(2)
1 + Q(4) + A2*C6-A3*S6))*(C8*(A2*C6-A3*S6)-S8*(-A2*S6-A3*C6))*2)*((
1D1P + A1BAR-G1)**2 + (-D2P + C8*(-G2-Q(2) + Q(4) + A2*C6-A3*S6) +
```

```
1S8*(-G3 + Q(5) + A2*S6 + A3*C6))**2 + (-D3P + C8*(-G3 + Q(5) + A2*
1S6 + A3*C6)-S8*(-G2-Q(2) + Q(4) + A2*C6-A3*S6))**2)**(-.5 )*(((D1P
1 + A1BAR-G1)**2 + (-D2P-G2 + A2)**2 + (-D3P-G3 + A3)**2)**.5-((D1P
1 + A1BAR-G1)**2 + (-D2P + C8*(-G2-Q(2) + Q(4) + A2*C6-A3*S6) + S8*
1(-G3 + Q(5) + A2*S6 + A3*C6))**2 + (-D3P + C8*(-G3 + Q(5) + A2*S6
1+ A3*C6)-S8*(-G2-Q(2) + Q(4) + A2*C6-A3*S6))**2)**.5)*.5
 PU( 3, 8,12, 2) = -XKK2*((-D2P + C8*(-G2-Q(2) + Q(4) + A2*C6-A3*S6
1) + S8*(-G3 + Q(5) + A2*S6 + A3*C6))*(C8*(-G3 + Q(5) + A2*S6 + A3*
1C6)-S8*(-G2-Q(2) + Q(4) + A2*C6-A3*S6))*2 + (-D3P + C8*(-G3 + Q(5)
1 + A2*S6 + A3*C6)-S8*(-G2-Q(2) + Q(4) + A2*C6-A3*S6))*(-C8*(-G2-Q(
12) + Q(4) + A2*C6-A3*S6)-S8*(-G3 + Q(5) + A2*S6 + A3*C6))*2)*((D1P
1 + A1BAR-G1)**2 + (-D2P + C8*(-G2-Q(2) + Q(4) + A2*C6-A3*S6) + S8*
1(-G3 + Q(5) + A2*S6 + A3*C6))**2 + (-D3P + C8*(-G3 + Q(5) + A2*S6
1+ A3*C6)-S8*(-G2-Q(2) + Q(4) + A2*C6-A3*S6))**2)**(-.5 )*(((D1P +
1A1BAR-G1)**2 + (-D2P-G2 + A2)**2 + (-D3P-G3 + A3)**2)**.5-((D1P +
1A1BAR-G1)**2 + (-D2P + C8*(-G2-Q(2) + Q(4) + A2*C6-A3*S6) + S8*(-G
13 + Q(5) + A2*S6 + A3*C6))**2 + (-D3P + C8*(-G3 + Q(5) + A2*S6 + A
13*C6)-S8*(-G2-Q(2) + Q(4) + A2*C6-A3*S6))**2)**.5)*.5


      53
READY

 FORMAC.CARDS.DATA

  PU( 3, 2,12, 2) = -ZZ(59)*(-ZZ(45)*C8*2 + ZZ(46)*S8*2)*.5
  PU( 3, 4,12, 2) = -ZZ(59)*(ZZ(45)*C8*2-ZZ(46)*S8*2)*.5
  PU( 3, 5,12, 2) = -ZZ(59)*(ZZ(45)*S8*2 + ZZ(46)*C8*2)*.5
  PU( 3, 6,12, 2) = -ZZ(59)*(ZZ(45)*(ZZ(41)*S8 + C8*(-A2*S6-A3*C6))*
12 + ZZ(46)*(ZZ(41)*C8-S8*(-A2*S6-A3*C6))*2)*.5
  PU( 3, 8,12, 2) = -ZZ(59)*(ZZ(45)*(ZZ(43)*C8-ZZ(44)*S8)*2 + ZZ(46)
1*(-ZZ(43)*S8-ZZ(44)*C8)*2)*.5


      7
READY
```

APPENDIX F

FORTRAN Program and Sample Data

APPENDIX F

FORTRAN Program and Sample Data

This appendix contains a listing of the FORTRAN program and a sample of the output data. The correspondence between the column headings of the output and the Q(i), QD(i), QDD(i) along with the units can be obtained in section 3.1. All units are in inch, pound, seconds, and radians.

```
C
C     ELEVEN DOF MODEL OF M110    TOM STREETER JAN 1977
C
C
C     THE MAIN PROGRAM ESSENTIALLY READS AND WRITES DATA.  CONSTANTS
C     ARE INITIALIZED IN DERIV AND SUBROUTINE KUTTA INTEGRATES THE
C     DIFFERENTIAL EQUATIONS.
C
      COMMON /DATA1/TIME,TIMEH,TIMEH2,TIMEH8
      COMMON /DERIV1/Q(11),QD(11),QDD(11)
      COMMON /KUTTA3/IBPTS,IRPTS,IGPTS
      COMMON /XREAD1/BRCHX(105),BRCHY(105),RODX(105),RODY(105),
     1 GAMMAX(105),GAMMAY(105)
      COMMON /NAME9/HOUR1
      READ 1,IBPTS,(BRCHX(I),BRCHY(I),I=1,IBPTS)
      READ 1,IRPTS,(RODX(I),RODY(I),I=1,IRPTS)
      READ 1,IGPTS,(GAMMAX(I),GAMMAY(I),I=1,IGPTS)
    1 FORMAT(I10/(8F10.0))
      DO 15 I=1,IGPTS
   15 GAMMAX(I)=GAMMAX(I)*3.14159/180.
      FINI=.005
      HOOR1=1.
      HOOR1=0.
      TIMEH2=TIMEH/2.
      TIMEH8=TIMEH/8.
      PRINT 2
    2 FORMAT(6X,4HTIME/)
      PRINT 3
    3 FORMAT(6X,3HETA,11X,1HV,11X,1HX,11X,1HY,11X,1HZ, 9X,3HPHI,
     1  9X,3HGAM, 9X,2HNU, 9X,4HTHET, 9X,3HPSI, 9X,3HTAU)
C
C     INITIALIZE CONSTANTS IN SUBROUTINE DERIV
C
      CALL DERIV
   20 CALL KUTTA
      PRINT 10,TIME
      PRINT 10,(Q(I),I=1,11)
      PRINT 10,(QD(I),I=1,11)
      PRINT 10,(QDD(I),I=1,11)
   10 FORMAT(11F12.4)
      PRINT 11
   11 FORMAT(//)
      IF(TIME .LE. FINI) GO TO 20
      STOP
      END


      SUBROUTINE DERIV
C
C     THIS SUBROUTINE INITIALIZES CONSTANTS AND DEFINES THE IMBEDDED
C     TERMS, ZZ(I), WHICH WERE REMOVED FROM THE PARTIAL DERIVITIVES.  THESE
C     TERMS ARE USED IN DER1 AND DER2.
C
      COMMON /DERIV1/Q(11),QD(11),QDD(11)
      COMMON /DERIV2/A1,A2,A3,ASTAR,XKY1,XKY2
      COMMON /DERIV3/A1SUB,A2SUB,A3SUB,XKK1,XKK2,A1BAR
      COMMON /DERIV4/XL(2,2),XM(2,2),XN(2,2),XK(2,2)
      COMMON /DERIV5/E1,E2,E3,FF1,FF2,FF3,D1,D2,D3,XI1,ETA1,ZETA1
      COMMON /DERIV6/B1,B2,B3,B1BAR,XI,ZETA
      COMMON /DERIV7/PG(6,12,12,3)
      COMMON /DERIV8/XIB,EB,ZETAB,XIR,ER,ZETAR,XIC,EC,ZETAC
      COMMON /DERIV9/BLFT,COFT,RDFT,FOFG
      COMMON /NAME1/PT(8,11,12,3),PW(5,12,12,3),PU(4,11,12,4)
     1 ,PD(3,11,12,4)
      COMMON /NAME7/DA(1,1,1,4),DP(1,1,1,4)
      COMMON /XZZ90/ZZ(90)
      COMMON /DERCON/     G1,G2,G3,D1P,D2P,D3P,C1P,C2P,C3P,ALPHA1,
     1 ALPHA2,ALPHA3,HH1,HH2,HH3
      COMMON /TRIG/C6,S6,C7,S7,C8,S8,C9,S9,C10,S10,C11,S11
      ASTAR=ASTAR*3.14159/180.
      G1=0.
      G2=-125.1 - 54.1931*SIN(ASTAR)
      G3=-7.15 - 54.1931*COS(ASTAR)
      D1P=39.5
      D2P=54.1931*SIN(ASTAR) - 54.0327*SIN(ASTAR + 0.2626)
      D3P=54.1931*COS(ASTAR) - 54.0327*COS(ASTAR + 0.2626)
      C1P=D1P
      C2P=D2P
      C3P=D3P
      ALPHA1=39.5
      ALPHA2=-125.1 - 54.0327*SIN(ASTAR + 0.2626)
      ALPHA3=-7.15 - 54.0327*COS(ASTAR + 0.2626)
      HH1=0.
      HH2=54.1931*SIN(ASTAR) - 2.*54.0327/3.*SIN(ASTAR + 0.2626)
      HH3=54.1931*COS(ASTAR) - 2.*54.0327/3.*COS(ASTAR + 0.2626)
      RETURN
      ENTRY DERFUC
C
C     ZERO ALL DERIVATIVE FUNCTIONS W/R TO KINETIC ENERGY
C     (NOT INCLUDING THE ANGULAR TERMS)
C
      DO 1 I=1,8
      DO 1 J=1,11
      DO 1 K=1,12
      DO 1 L=1,3
    1 PT(I,J,K,L)=0.
C
C     ZERO KINETIC ENERGY ANGULAR TERMS
C
      DO 12 I=1,5
```

```
      DO 12 J=1,12
      DO 12 K=1,12
      DO 12 L=1,3
   12 PW(1,J,K,L)=0.
C
C     ZERO DERIVATIVE FUNCTIONS W/R TO POTENTIAL ENERGY  U2
C
      DO 4 I=1,4
      DO 4 J=1,11
      DO 4 K=1,12
      DO 4 L=1,4
    4 PU(I,J,K,L)=0.
C
C     ZERO DERIVATIVE FUNCTIONS W/R TO DISSIPATIVE ENERGY
C
      DO 5 I=1,3
      DO 5 J=1,11
      DO 5 K=1,12
      DO 5 L=1,4
    5 PD(I,J,K,L)=0.
C
C     ZERO DERIVATIVE FUNCTIONS W/R TO GENERALIZED FORCES
C
      DO 6 I=1,6
      DO 6 J=1,12
      DO 6 K=1,12
      DO 6 L=1,3
    6 PG(I,J,K,L)=0.
      C6=COS(Q(6))
      S6=SIN(Q(6))
      C7=COS(Q(7))
      S7=SIN(Q(7))
      C8=COS(Q(8))
      S8=SIN(Q(8))
      C9=COS(Q(9))
      S9=SIN(Q(9))
      C10=COS(Q(10))
      S10=SIN(Q(10))
      C11=COS(Q(11))
      S11=SIN(Q(11))
      ZZ(1)=C8*S9 + S8*C9*S10
      ZZ(2)=-C8*S9 - S8*C9*S10
      ZZ(3)=C8*C9*S10 - S8*S9
      ZZ(4)=-C8*C9*S10 + S8*S9
      ZZ(5)=C8*S9*S10 + S8*C9
      ZZ(6)=-C8*S9*S10 - S8*C9
      ZZ(7)=C8*C9 - S8*S9*S10
      ZZ(8)=-C8*C9 + S8*S9*S10
      ZZ(9)=C9*S10*S6 + S9*C6
      ZZ(10)=-C9*S10*S6 - S9*C6
      ZZ(11)=C9*S10*C6 - S9*S6
      ZZ(12)=-C9*S10*C6 + S9*S6
      ZZ(13)=C9*S6 + S9*S10*C6
      ZZ(14)=-C9*S6 - S9*S10*C6
      ZZ(15)=C9*C6 - S9*S10*S6
      ZZ(16)=-C9*C6 + S9*S10*S6
      ZZ(17)=-C11*C9*S10 - S11*C9*C10*C6
      ZZ(18)=C11*C9*S10 + S11*C9*C10*C6
      ZZ(19)=-C11*C9*C10*C6 + S11*C9*S10
      ZZ(20)=C11*C9*C10*C6 - S11*C9*S10
      ZZ(21)=C11*C10 - S11*S10*C6
      ZZ(22)=-C11*C10 + S11*S10*C6
      ZZ(23)=-C11*S10*C6 - S11*C10
      ZZ(24)=C11*S10*C6 + S11*C10
      ZZ(25)=C11*C10*C6 - S11*S10
      ZZ(26)=-C11*C10*C6 + S11*S10
      ZZ(27)=-C11*S10 - S11*C10*C6
      ZZ(28)=C11*S10 + S11*C10*C6
      ZZ(29)=C11*S9*S10 + S11*S9*C10*C6
      ZZ(30)=-C11*S9*S10 - S11*S9*C10*C6
      ZZ(31)=C11*S9*C10*C6 - S11*S9*S10
      ZZ(32)=-C11*S9*C10*C6 + S11*S9*S10
      ZZ(33)=-C11*C9*C10 + S11*C9*S10*C6
      ZZ(34)=C11*C9*C10 - S11*C9*S10*C6
      ZZ(35)=C11*C9*S10*C6 + S11*C9*C10
      ZZ(36)=-C11*C9*S10*C6 - S11*C9*C10
      ZZ(37)=C11*S9*C10 - S11*S9*S10*C6
      ZZ(38)=-C11*S9*C10 + S11*S9*S10*C6
      ZZ(39)=-C11*S9*S10*C6 - S11*S9*C10
      ZZ(40)=C11*S9*S10*C6 + S11*S9*C10
      ZZ(41)=A2SUB*C6 - A3SUB*S6
      ZZ(42)=A2SUB*S6 + A3SUB*C6
      ZZ(43)=-G3 + Q(5) + ZZ(42)
      ZZ(44)=-G2 - Q(2) + Q(4) + ZZ(41)
      ZZ(45)=-D2P + ZZ(43)*S8 + ZZ(44)*C8
      ZZ(46)=-D3P + ZZ(43)*C8 - ZZ(44)*S8
      ZZ(47)=-D1P - G1 + A1SUB
      ZZ(48)=-D2P - G2 + A2SUB
      ZZ(49)=-D3P + A3SUB - G3
      ZZ(50)=ZZ(47)**2 + ZZ(45)**2 + ZZ(46)**2
      ZZ(51)=ZZ(47)**2 + ZZ(48)**2 + ZZ(49)**2
      ZZ(52)=-ZZ(50)**0.5 + ZZ(51)**0.5
      ZZ(53)=ZZ(52)*XKK1*ZZ(50)**(-.5)
      ZZ(54)=D1P + A1BAR - G1
      ZZ(55)=-G2 - D2P + A2SUB
      ZZ(56)=ZZ(54)**2 + ZZ(45)**2 + ZZ(46)**2
      ZZ(57)=ZZ(54)**2 + ZZ(55)**2 + ZZ(49)**2
      ZZ(58)=-ZZ(56)**0.5 + ZZ(57)**0.5
      ZZ(59)=XKK2*ZZ(58)*ZZ(56)**(-.5)
      ZZ(60)=-XN(1,1) - A3 + XN(1,1)*ZZ(15) - XL(1,1)*S9*C10 +
     1 XM(1,1)*ZZ(13) + A2*S9*S10 + A3*C9 + Q(4)*S9*S10 + Q(5)*C9 -
     2 A1*S9*C10
      ZZ(61)=-XN(1,2) - A3 - XL(1,2)*S9*C10 + XM(1,2)*ZZ(13) +
     1 XN(1,2)*ZZ(15) + Q(4)*S9*S10 + Q(5)*C9 - A1*S9*C10 + A3*C9
     2 + A2*S9*S10
      ZZ(62)=-XN(2,1) - A3 - XL(2,1)*S9*C10 + XM(2,1)*ZZ(13) +
```

```
     1 XN(2,1)*ZZ(15) + Q(4)*S9*S10 + Q(5)*C9 - A1*S9*C10 + A3*C9 +
     2 A2*S9*S10
      ZZ(63)=-XN(2,2) - A3 + XM(2,2)*ZZ(13) + XN(2,2)*ZZ(15) -
     1 XL(2,2)*S9*C10 + Q(4)*S9*S10 + Q(5)*C9 - A1*S9*C10 + A3*C9 +
     2 A2*S9*S10
      ZZ(64)=-ALPHA2 + G2 + Q(2) + A2 + C8*(-G2 - Q(2) - A2 +
     1 ALPHA2*C10 + ALPHA3*S9*S10 - C9*S10*(ALPHA1 - Q(3))) +
     2 S8*(-G3 - A3 + ALPHA3*C9 + S9*(ALPHA1 - Q(3)))
      ZZ(65)=-ALPHA2 + G2 + Q(2) + A2 + C8*(-G2 - Q(2) - A2 +
     1 ALPHA2*C10 + ALPHA3*S9*S10 - C9*S10*(-ALPHA1 - Q(3))) +
     2 S8*(-A3 - G3 + ALPHA3*C9 + S9*(-ALPHA1 - Q(3)))
      ZZ(66)=-G2 - Q(2) + Q(4) + B2*C6 - B3*S6
      ZZ(67)=-G3 + Q(5) + B2*S6 + B3*C6
      ZZ(68)=-C1P + B1 - G1
      ZZ(69)=-C2P + ZZ(67)*S8 + ZZ(66)*C8
      ZZ(70)=-C3P + ZZ(67)*C8 - ZZ(66)*S8
      ZZ(71)=(ZZ(70)**2 + ZZ(68)**2 + ZZ(69)**2)**(-.5)*0.5
      ZZ(72)=-B2*S6 - B3*C6
      ZZ(73)=B2*C6 - B3*S6
      ZZ(74)=(ZZ(70)**2 + ZZ(69)**2 + (B1BAR + C1P - G1)**2)**(-.5)*0.5
      ZZ(75)=-G2 - Q(2) + Q(4) - A3SUB*S6 + A2SUB*C6
      ZZ(76)=-G3 + Q(5) + A3SUB*C6 + A2SUB*S6
      ZZ(77)=-D2P + ZZ(76)*S8 + ZZ(75)*C8
      ZZ(78)=-D3P + ZZ(76)*C8 - ZZ(75)*S8
      ZZ(79)=-A3SUB*S6 + A2SUB*C6
      ZZ(80)=-A3SUB*C6 - A2SUB*S6
      ZZ(81)=(ZZ(77)**2 + ZZ(78)**2 + (D1P + A1BAR - G1)**2)**(-.5)*0.5
      ZZ(82)=(ZZ(77)**2 + ZZ(78)**2 + (A1SUB - D1P - G1)**2)**(-.5)*0.5
      ZZ(83)=A2*S9*S10 + A3*C9 + Q(4)*S9*S10 + Q(5)*C9 - A1*S9*C10
      ZZ(84)=-A2*C9*S10 + A3*S9 - Q(4)*C9*S10 + Q(5)*S9 + A1*C9*C10
      ZZ(85)=ZZ(12)*C11 - C9*C10*S11
      ZZ(86)=-C9*C10*C11*C6 + C9*S10*S11
      ZZ(87)=C10*C11*C6 - S10*S11
      ZZ(88)=-C10*S11*C6-S10*C11
      ZZ(89)=S9*C10*C11*C6 - S9*S10*S11
      ZZ(90)=ZZ(13)*C11 + S9*C10*S11
      CALL DER1
      CALL DER2
      RETURN
      END


      SUBROUTINE DER1
C
C     THIS SUBROUTINE DEFINES THE PARTIAL DERIVATIVES PERTAINING TO THE
C     TRANSLATIONAL PART OF THE KINETIC ENERGY.
C
      COMMON /DERIV1/Q(11),QD(11),QDD(11)
      COMMON /DERIV2/A1,A2,A3,ASTAR,XKY1,XKY2
      COMMON /DERIV3/A1SUB,A2SUB,A3SUB,XKK1,XKK2,A1BAR
      COMMON /DERIV4/XL(2,2),XM(2,2),XN(2,2),XK(2,2)
      COMMON /DERIV5/E1,E2,E3,FF1,FF2,FF3,D1,D2,D3,XI1,ETA1,ZETA1
      COMMON /DERIV6/B1,B2,B3,B1BAR,XI,ZETA
      COMMON /DERIV7/PG(6,12,12,3)
      COMMON /DERIV8/XIB,EB,ZETAB,XIR,ER,ZETAR,XIC,EC,ZETAC
      COMMON /DERIV9/BLFT,CUFT,ROFT,FUFG
      COMMON /NAME1/PT(8,11,12,3),PW(5,12,12,3),PU(4,11,12,4)
     1 ,PD(3,11,12,4)
      COMMON /NAME7/DA(1,1,1,4),DP(1,1,1,4)
      COMMON /XZZ90/ZZ(90)
      COMMON /DERCON/     G1,G2,G3,D1P,D2P,D3P,C1P,C2P,C3P,ALPHA1,
     1 ALPHA2,ALPHA3,HH1,HH2,HH3
      COMMON /TRIG/C6,S6,C7,S7,C8,S8,C9,S9,C10,S10,C11,S11
C
C     DERIVATIVE FUNCTIONS W/R TO K.E. (NOT INCLUDING ANGULAR TERMS)
C
      PT( 1, 3,12, 1) = 1
      PT( 1, 9,12, 1) = A2*S9*S10 + A3*C9-A1*C10*S9
      PT( 1,10,12, 1) = -A2*C10*C9-A1*C9*S10
      PT( 1,10,12, 2) = -A2*S10 + A1*C10
      PT( 1,10,12, 2) = -A2*S10 + A1*C10
      PT( 1, 9,12, 3) = A2*C9*S10-A3*S9-A1*C10*C9
      PT( 1,10,12, 3) = A2*C10*S9 + A1*S9*S10
      PT( 2, 4,12, 1) = -C9*S10
      PT( 2, 5,12, 1) = S9
      PT( 2, 9,12, 1) = Q(4)*S9*S10 + Q(5)*C9
      PT( 2,10,12, 1) = -Q(4)*C10*C9
      PT( 2, 4,12, 2) = C10
      PT( 2,10,12, 2) = -Q(4)*S10
      PT( 2, 4,12, 3) = S9*S10
      PT( 2, 5,12, 3) = C9
      PT( 2, 9,12, 3) = Q(4)*C9*S10-Q(5)*S9
      PT( 2,10,12, 3) = Q(4)*C10*S9
      PT( 3, 2,12, 1) = -C9*S10
      PT( 3, 8,12, 1) = HH2*ZZ(1) + HH3*ZZ(3)
      PT( 3, 9,12, 1) = -HH1*C10*S9 + HH2*ZZ(5) + HH3*ZZ(7) + G3*C9-G1*C
     110*S9 + S9*S10*(G2 + Q(2))
      PT( 3,10,12, 1) = -HH1*C9*S10-HH2*C10*C8*C9 + HH3*C10*S8*C9-G1*C9*
     1S10-C10*C9*(G2 + Q(2))
      PT( 3, 2,12, 2) = C10
      PT( 3, 8,12, 2) = -HH2*C10*S8-HH3*C10*C8
      PT( 3,10,12, 2) = HH1*C10-HH2*C8*S10 + HH3*S8*S10 + G1*C10-S10*(G2
     1 + Q(2))
      PT( 3, 2,12, 3) = S9*S10
      PT( 3, 8,12, 3) = HH2*ZZ(7) + HH3*ZZ(6)
```

```
      PT( 3, 9,12, 3) = -HH1*C10*C9 + HH2*ZZ(3) + HH3*ZZ(2)-G3*S9-G1*C10
     1*C9 + C9*S10*(G2 + Q(2))
      PT( 3,10,12, 3) = HH1*S9*S10 + HH2*C10*C8*S9-HH3*C10*S9*S8 + G1*S9
     1*S10 + C10*S9*(G2 + Q(2))
      PT( 1, 9, 9, 1) = A2*C9*S10-A3*S9-A1*C10*C9
      PT( 1, 9,10, 1) = A2*C10*S9 + A1*S9*S10
      PT( 1,10,10, 1) = A2*C9*S10-A1*C10*C9
      PT( 1,10,10, 2) = -A2*C10-A1*S10
      PT( 1, 9, 9, 3) = -A2*S9*S10-A3*C9 + A1*C10*S9
      PT( 1, 9,10, 3) = A2*C10*C9 + A1*C9*S10
      PT( 1,10,10, 3) = -A2*S9*S10 + A1*C10*S9
      PT( 2, 4, 9, 1) = S9*S10
      PT( 2, 4,10, 1) = -C10*C9
      PT( 2, 5, 9, 1) = C9
      PT( 2, 9, 9, 1) = Q(4)*C9*S10-Q(5)*S9
      PT( 2, 9,10, 1) = Q(4)*C10*S9
      PT( 2,10,10, 1) = Q(4)*C9*S10
      PT( 2, 4,10, 2) = -S10
      PT( 2,10,10, 2) = -Q(4)*C10
      PT( 2, 4, 9, 3) = C9*S10
      PT( 2, 4,10, 3) = C10*S9
      PT( 2, 5, 9, 3) = -S9
      PT( 2, 5, 9, 3) = -S9
      PT( 2, 9, 9, 3) = -Q(4)*S9*S10-Q(5)*C9
      PT( 2, 9,10, 3) = Q(4)*C10*C9
      PT( 2, 9,10, 3) = Q(4)*C10*C9
      PT( 2,10,10, 3) = -Q(4)*S9*S10
      PT( 3, 2, 9, 1) = S9*S10
      PT( 3, 2,10, 1) = -C10*C9
      PT( 3, 8, 8, 1) = HH2*ZZ(3) + HH3*ZZ(2)
      PT( 3, 8, 9, 1) = HH2*ZZ(7) + HH3*ZZ(6)
      PT( 3, 8,10, 1) = HH2*C10*S8*C9 + HH3*C10*C8*C9
      PT( 3, 9, 9, 1) = -HH1*C10*C9 + HH2*ZZ(3) + HH3*ZZ(2)-G3*S9-G1*C10
     1*C9 + C9*S10*(G2 + Q(2))
      PT( 3, 9,10, 1) = HH1*S9*S10 + HH2*C10*C8*S9-HH3*C10*S9*S8 + G1*S9
     1*S10 + C10*S9*(G2 + Q(2))
      PT( 3,10,10, 1) = -HH1*C10*C9 + HH2*C8*C9*S10-HH3*S8*C9*S10-G1*C10
     1*C9 + C9*S10*(G2 + Q(2))
      PT( 3, 2,10, 2) = -S10
      PT( 3, 8, 8, 2) = -HH2*C10*C8 + HH3*C10*S8
      PT( 3, 8,10, 2) = HH2*S8*S10 + HH3*C8*S10
      PT( 3,10,10, 2) = -HH1*S10-HH2*C10*C8 + HH3*C10*S8-G1*S10-C10*(G2
     1+ Q(2))
      PT( 3, 2, 9, 3) = C9*S10
      PT( 3, 2,10, 3) = C10*S9
      PT( 3, 8, 8, 3) = HH2*ZZ(6) + HH3*ZZ(8)
      PT( 3, 8, 9, 3) = HH2*ZZ(2) + HH3*ZZ(4)
      PT( 3, 8,10, 3) = -HH2*C10*S9*S8-HH3*C10*C8*S9
      PT( 3, 9, 9, 3) = HH1*C10*S9 + HH2*ZZ(6) + HH3*ZZ(8)-G3*C9 + G1*C1
     10*S9-S9*S10*(G2 + Q(2))
      PT( 3, 9,10, 3) = HH1*C9*S10 + HH2*C10*C8*C9-HH3*C10*S8*C9 + G1*C9
     1*S10 + C10*C9*(G2 + Q(2))
      PT( 3,10,10, 3) = HH1*C10*S9-HH2*C8*S9*S10 + HH3*S9*S8*S10 + G1*C1

     10*S9-S9*S10*(G2 + Q(2))
      PT( 4, 6,12, 1) = E2*ZZ(9) + E3*ZZ(11)
      PT( 4, 9,12, 1) = E2*ZZ(13) + E3*ZZ(15)-E1*C10*S9
      PT( 4,10,12, 1) = -E2*C10*C6*C9 + E3*C10*S6*C9-E1*C9*S10
      PT( 4, 6,12, 2) = -E2*C10*S6-E3*C10*C6
      PT( 4,10,12, 2) = -E2*C6*S10 + E3*S6*S10 + E1*C10
      PT( 4, 6,12, 3) = E2*ZZ(15) + E3*ZZ(14)
      PT( 4, 9,12, 3) = E2*ZZ(11) + E3*ZZ(10)-E1*C10*C9
      PT( 4,10,12, 3) = E2*C10*C6*S9-E3*C10*S6*S9 + E1*S9*S10
      PT( 4, 6, 6, 1) = E2*ZZ(11) + E3*ZZ(10)
      PT( 4, 6, 9, 1) = E2*ZZ(15) + E3*ZZ(14)
      PT( 4, 6,10, 1) = E2*C10*S6*C9 + E3*C10*C6*C9
      PT( 4, 6,10, 1) = E2*C10*S6*C9 + E3*C10*C6*C9
      PT( 4, 9, 9, 1) = E2*ZZ(11) + E3*ZZ(10)-E1*C10*C9
      PT( 4, 9,10, 1) = E2*C10*C6*S9-E3*C10*S6*S9 + E1*S9*S10
      PT( 4,10,10, 1) = E2*C6*C9*S10-E3*S6*C9*S10-E1*C10*C9
      PT( 4, 6, 6, 2) = -E2*C10*C6 + E3*C10*S6
      PT( 4, 6,10, 2) = E2*S6*S10 + E3*C6*S10
      PT( 4,10,10, 2) = -E2*C10*C6 + E3*C10*S6-E1*S10
      PT( 4, 6, 6, 3) = E2*ZZ(14) + E3*ZZ(16)
      PT( 4, 6, 9, 3) = E2*ZZ(10) + E3*ZZ(12)
      PT( 4, 6,10, 3) = -E2*C10*S6*S9-E3*C10*C6*S9
      PT( 4, 9, 9, 3) = E2*ZZ(14) + E3*ZZ(16) + E1*C10*S9
      PT( 4, 9,10, 3) = E2*C10*C6*C9-E3*C10*S6*C9 + E1*C9*S10
      PT( 4,10,10, 3) = -E2*C6*S9*S10 + E3*S6*S9*S10 + E1*C10*S9
      PT( 5, 6,12, 1) = FF2*C11*ZZ(9) + FF3*ZZ(11) + FF1*S11*ZZ(9)
      PT( 5, 9,12, 1) = FF2*(C11*ZZ(13) + S11*C10*S9) + FF3*ZZ(15) + FF1
     1*(-C11*C10*S9 + S11*ZZ(13))
      PT( 5,10,12, 1) = FF2*ZZ(19) + FF3*C10*S6*C9 + FF1*ZZ(17)
      PT( 5,11,12, 1) = FF2*(-C11*C10*C9-S11*ZZ(12)) + FF1*(C11*ZZ(12)-S
     111*C10*C9)
      PT( 5, 6,12, 2) = -FF2*C11*C10*S6-FF3*C10*C6-FF1*S11*C10*S6
      PT( 5,10,12, 2) = FF2*ZZ(23) + FF3*S6*S10 + FF1*ZZ(21)
      PT( 5,11,12, 2) = FF2*ZZ(27) + FF1*ZZ(25)
      PT( 5, 6,12, 3) = FF2*C11*ZZ(15) + FF3*ZZ(14) + FF1*S11*ZZ(15)
      PT( 5, 9,12, 3) = FF2*(C11*ZZ(11) + S11*C10*C9) + FF3*ZZ(10) + FF1
     1*(-C11*C10*C9 + S11*ZZ(11))
      PT( 5,10,12, 3) = FF2*ZZ(31)-FF3*C10*S6*S9 + FF1*ZZ(29)
      PT( 5,11,12, 3) = FF2*(C11*C10*S9-S11*ZZ(13)) + FF1*(C11*ZZ(13) +
     1S11*C10*S9)
      PT( 5, 6, 6, 1) = FF2*C11*ZZ(11) + FF3*ZZ(10) + FF1*S11*ZZ(11)
      PT( 5, 6, 9, 1) = FF2*C11*ZZ(15) + FF3*ZZ(14) + FF1*S11*ZZ(15)
      PT( 5, 6,10, 1) = FF2*C11*C10*S6*C9 + FF3*C10*C6*C9 + FF1*S11*C10*
     1S6*C9
      PT( 5, 6,11, 1) = -FF2*S11*ZZ(9) + FF1*C11*ZZ(9)
      PT( 5, 9, 9, 1) = FF2*(C11*ZZ(11) + S11*C10*C9) + FF3*ZZ(10) + FF1
     1*(-C11*C10*C9 + S11*ZZ(11))
      PT( 5, 9,10, 1) = FF2*ZZ(31)-FF3*C10*S6*S9 + FF1*ZZ(29)
      PT( 5, 9,11, 1) = FF2*(C11*C10*S9-S11*ZZ(13)) + FF1*(C11*ZZ(13) +
     1S11*C10*S9)
      PT( 5,10,10, 1) = FF2*ZZ(35)-FF3*S6*C9*S10 + FF1*ZZ(33)
      PT( 5,10,11, 1) = FF2*ZZ(18) + FF1*ZZ(19)
      PT( 5,11,11, 1) = FF2*(-C11*ZZ(12) + S11*C10*C9) + FF1*(-C11*C10*C
```

```
                 19-S11*ZZ(12))
0132              PT( 5, 6, 6, 2) = -FF2*C11*C10*C6 + FF3*C10*S6-FF1*S11*C10*C6
0133              PT( 5, 6,10, 2) = FF2*C11*S6*S10 + FF3*C6*S10 + FF1*S11*S6*S10
0134              PT( 5, 6,11, 2) = FF2*S11*C10*S6-FF1*C11*C10*S6
0135              PT( 5,10,10, 2) = FF2*ZZ(26) + FF3*C10*S6 + FF1*ZZ(27)
0136              PT( 5,10,11, 2) = FF2*ZZ(22) + FF1*ZZ(23)
0137              PT( 5,11,11, 2) = FF2*ZZ(26) + FF1*ZZ(27)
0138              PT( 5, 6, 6, 3) = FF2*C11*ZZ(14) + FF3*ZZ(16) + FF1*S11*ZZ(14)
0139              PT( 5, 6, 9, 3) = FF2*C11*ZZ(10) + FF3*ZZ(12) + FF1*S11*ZZ(10)
0140              PT( 5, 6,10, 3) = -FF2*C11*C10*S6*S9-FF3*C10*C6*S9-FF1*S11*C10*S6*
                 1S9
0141              PT( 5, 6,11, 3) = -FF2*S11*ZZ(15) + FF1*C11*ZZ(15)
0142              PT( 5, 9, 9, 3) = FF2*(C11*ZZ(14)-S11*C10*S9) + FF3*ZZ(16) + FF1*(
                 1C11*C10*S9 + S11*ZZ(14))
0143              PT( 5, 9,10, 3) = FF2*ZZ(20)-FF3*C10*S6*C9 + FF1*ZZ(18)
0144              PT( 5, 9,11, 3) = FF2*(C11*C10*C9-S11*ZZ(11)) + FF1*(C11*ZZ(11) +
                 1S11*C10*C9)
0145              PT( 5,10,10, 3) = FF2*ZZ(39) + FF3*S6*S9*S10 + FF1*ZZ(37)
0146              PT( 5,10,11, 3) = FF2*ZZ(30) + FF1*ZZ(31)
0147              PT( 5,11,11, 3) = FF2*(-C11*ZZ(13)-S11*C10*S9) + FF1*(C11*C10*S9-S
                 111*ZZ(13))
0148              PT( 6, 6,12, 1) = D3*ZZ(11) + D1*S11*ZZ(9) + D2*C11*ZZ(9)
0149              PT( 6, 9,12, 1) = D3*ZZ(15) + D1*(-C11*C10*S9 + S11*ZZ(13)) + D2*(
                 1C11*ZZ(13) + S11*C10*S9)
0150              PT( 6,10,12, 1) = D3*C10*S6*C9 + D1*ZZ(17) + D2*ZZ(19)
0151              PT( 6,11,12, 1) = D1*(C11*ZZ(12)-S11*C10*C9) + D2*(-C11*C10*C9-S11
                 1*ZZ(12))
0152              PT( 6, 6,12, 2) = -D3*C10*C6-D1*S11*C10*S6-D2*C11*C10*S6
0153              PT( 6,10,12, 2) = D3*S6*S10 + D1*ZZ(21) + D2*ZZ(23)
0154              PT( 6,11,12, 2) = D1*ZZ(25) + D2*ZZ(27)
0155              PT( 6, 6,12, 3) = D3*ZZ(14) + D1*S11*ZZ(15) + D2*C11*ZZ(15)
0156              PT( 6, 9,12, 3) = D3*ZZ(10) + D1*(-C11*C10*C9 + S11*ZZ(11)) + D2*(
                 1C11*ZZ(11) + S11*C10*C9)
0157              PT( 6,10,12, 3) = -D3*C10*S6*S9 + D1*ZZ(29) + D2*ZZ(31)
0158              PT( 6,11,12, 3) = D1*(C11*ZZ(13) + S11*C10*S9) + D2*(C11*C10*S9-S1
                 11*ZZ(13))
0159              PT( 6, 6, 6, 1) = D3*ZZ(10) + D1*S11*ZZ(11) + D2*C11*ZZ(11)
0160              PT( 6, 6, 9, 1) = D3*ZZ(14) + D1*S11*ZZ(15) + D2*C11*ZZ(15)
0161              PT( 6, 6,10, 1) = D3*C10*C6*C9 + D1*S11*C10*S6*C9 + D2*C11*C10*S6*
                 1C9
0162              PT( 6, 6,11, 1) = D1*C11*ZZ(9)-D2*S11*ZZ(9)
0163              PT( 6, 9, 9, 1) = D3*ZZ(10) + D1*(-C11*C10*C9 + S11*ZZ(11)) + D2*(
                 1C11*ZZ(11) + S11*C10*C9)
0164              PT( 6, 9,10, 1) = -D3*C10*S6*S9 + D1*ZZ(29) + D2*ZZ(31)
0165              PT( 6, 9,11, 1) = D1*(C11*ZZ(13) + S11*C10*S9) + D2*(C11*C10*S9-S1
                 11*ZZ(13))
0166              PT( 6,10,10, 1) = -D3*S6*C9*S10 + D1*ZZ(33) + D2*ZZ(35)
0167              PT( 6,10,11, 1) = D1*ZZ(19) + D2*ZZ(18)
0168              PT( 6,11,11, 1) = D1*(-C11*C10*C9-S11*ZZ(12)) + D2*(-C11*ZZ(12) +
                 1S11*C10*C9)
0169              PT( 6, 6, 6, 2) = D3*C10*S6-D1*S11*C10*C6-D2*C11*C10*C6
0170              PT( 6, 6,10, 2) = D3*C6*S10 + D1*S11*S6*S10 + D2*C11*S6*S10
0171              PT( 6, 6,11, 2) = -D1*C11*C10*S6 + D2*S11*C10*S6

0172              PT( 6,10,10, 2) = D3*C10*S6 + D1*ZZ(27) + D2*ZZ(26)
0173              PT( 6,10,11, 2) = D1*ZZ(23) + D2*ZZ(22)
0174              PT( 6,11,11, 2) = D1*ZZ(27) + D2*ZZ(26)
0175              PT( 6, 6, 6, 3) = D3*ZZ(16) + D1*S11*ZZ(14) + D2*C11*ZZ(14)
0176              PT( 6, 6, 9, 3) = D3*ZZ(12) + D1*S11*ZZ(10) + D2*C11*ZZ(10)
0177              PT( 6, 6,10, 3) = -D3*C10*C6*S9-D1*S11*C10*S6*S9-D2*C11*C10*S6*S9
0178              PT( 6, 6,11, 3) = D1*C11*ZZ(15)-D2*S11*ZZ(15)
0179              PT( 6, 9, 9, 3) = D3*ZZ(16) + D1*(C11*C10*S9 + S11*ZZ(14)) + D2*(C
                 111*ZZ(14)-S11*C10*S9)
0180              PT( 6, 9,10, 3) = -D3*C10*S6*C9 + D1*ZZ(18) + D2*ZZ(20)
0181              PT( 6, 9,11, 3) = D1*(C11*ZZ(11) + S11*C10*C9) + D2*(C11*C10*C9-S1
                 11*ZZ(11))
0182              PT( 6,10,10, 3) = D3*S6*S9*S10 + D1*ZZ(37) + D2*ZZ(39)
0183              PT( 6,10,11, 3) = D1*ZZ(31) + D2*ZZ(30)
0184              PT( 6,11,11, 3) = D1*(C11*C10*S9-S11*ZZ(13)) + D2*(-C11*ZZ(13)-S11
                 1*C10*S9)
0185              PT( 7, 6,12, 1) = XI1*S11*ZZ(9) + ETA1*(C7*C11*ZZ(9) + S7*ZZ(11))
                 1+ ZETA1*(C7*ZZ(11)-S7*C11*ZZ(9))
0186              PT( 7, 7,12, 1) = ETA1*(C7*ZZ(9)-S7*(C11*ZZ(12)-S11*C10*C9)) + ZET
                 1A1*(-C7*(C11*ZZ(12)-S11*C10*C9)-S7*ZZ(9))
0187              PT( 7, 9,12, 1) = XI1*(-C11*C10*S9 + S11*ZZ(13)) + ETA1*(C7*(C11*Z
                 1Z(13) + S11*C10*S9) + S7*ZZ(15)) + ZETA1*(C7*ZZ(15)-S7*(C11*ZZ(13)
                 1 + S11*C10*S9))
0188              PT( 7,10,12, 1) = XI1*ZZ(17) + ETA1*(C7*ZZ(19) + S7*C10*S6*C9) + Z
                 1ETA1*(C7*C10*S6*C9-S7*ZZ(19))
0189              PT( 7,11,12, 1) = XI1*(C11*ZZ(12)-S11*C10*C9) + ETA1*C7*(-C11*C10*
                 1C9-S11*ZZ(12))-ZETA1*S7*(-C11*C10*C9-S11*ZZ(12))
0190              PT( 7, 6,12, 2) = -XI1*S11*C10*S6 + ETA1*(-C7*C11*C10*S6-S7*C10*C6
                 1) + ZETA1*(-C7*C10*C6 + S7*C11*C10*S6)
0191              PT( 7, 7,12, 2) = ETA1*(-C7*C10*S6-S7*ZZ(25)) + ZETA1*(-C7*ZZ(25)
                 1+ S7*C10*S6)
0192              PT( 7,10,12, 2) = XI1*ZZ(21) + ETA1 (C7*ZZ(23) + S7*S6*S10) + ZETA
                 11*(C7*S6*S10-S7*ZZ(23))
0193              PT( 7,11,12, 2) = XI1*ZZ(25) + ETA1*C7*ZZ(27)-ZETA1*S7*ZZ(27)
0194              PT( 7, 6,12, 3) = XI1*S11*ZZ(15) + ETA1*(C7*C11*ZZ(15) + S7*ZZ(14)
                 1) + ZETA1*(C7*ZZ(14)-S7*C11*ZZ(15))
0195              PT( 7, 7,12, 3) = ETA1*(C7*ZZ(15)-S7*(C11*ZZ(13) + S11*C10*S9)) +
                 1ZETA1*(-C7*(C11*ZZ(13) + S11*C10*S9)-S7*ZZ(15))
0196              PT( 7, 9,12, 3) = XI1*(-C11*C10*C9 + S11*ZZ(11)) + ETA1*(C7*(C11*Z
                 1Z(11) + S11*C10*C9) + S7*ZZ(10)) + ZETA1*(C7*ZZ(10)-S7*(C11*ZZ(11)
                 1 + S11*C10*C9))
0197              PT( 7,10,12, 3) = XI1*ZZ(29) + ETA1*(C7*ZZ(31)-S7*C10*S6*S9) + ZET
                 1A1*(-C7*C10*S6*S9-S7*ZZ(31))
0198              PT( 7,11,12, 3) = XI1*(C11*ZZ(13) + S11*C10*S9) + ETA1*C7*(C11*C10
                 1*S9-S11*ZZ(13))-ZETA1*S7*(C11*C10*S9-S11*ZZ(13))
0199              PT( 7, 6, 6, 1) = XI1*S11*ZZ(11) + ETA1*(C7*C11*ZZ(11) + S7*ZZ(10)
                 1) + ZETA1*(C7*ZZ(10)-S7*C11*ZZ(11))
0200              PT( 7, 6, 7, 1) = ETA1*(C7*ZZ(11)-S7*C11*ZZ(9)) + ZETA1*(-C7*C11*Z
                 1Z(9)-S7*ZZ(11))
0201              PT( 7, 6, 9, 1) = XI1*S11*ZZ(15) + ETA1*(C7*C11*ZZ(15) + S7*ZZ(14)
                 1) + ZETA1*(C7*ZZ(14)-S7*C11*ZZ(15))
0202              PT( 7, 6,10, 1) = XI1*S11*C10*S6*C9 + ETA1*(C7*C11*C10*S6*C9 + S7*
                 1C10*C6*C9) + ZETA1*(C7*C10*C6*C9-S7*C11*C10*S6*C9)
```

```
0203       PT( 7, 6,11, 1) = XI1*C11*ZZ(9)-ETA1*(C7*S11*ZZ(9) + ZETA1*S7*S11*Z
          1Z(9)
0204       PT( 7, 7, 7, 1) = ETA1*(-C7*(C11*ZZ(12)-S11*C10*C9)-S7*ZZ(9)) + ZE
          1TA1*(-C7*ZZ(9) + S7*(C11*ZZ(12)-S11*C10*C9))
0205       PT( 7, 7, 9, 1) = ETA1*(C7*ZZ(15)-S7*(C11*ZZ(13) + S11*C10*S9)) +
          1ZETA1*(-C7*(C11*ZZ(13) + S11*C10*S9)-S7*ZZ(15))
0206       PT( 7, 7,10, 1) = ETA1*(C7*C10*S6*C9-S7*ZZ(19)) + ZETA1*(-C7*ZZ(19
          1)-S7*C10*S6*C9)
0207       PT( 7, 7,11, 1) = -ETA1*S7*(-C11*C10*C9-S11*ZZ(12))-ZETA1*C7*(-C11
          1*C10*C9-S11*ZZ(12))
0208       PT( 7, 9, 9, 1) = XI1*(-C11*C10*C9 + S11*ZZ(11)) + ETA1*(C7*(C11*Z
          1Z(11) + S11*C10*C9) + S7*ZZ(10)) + ZETA1*(C7*ZZ(10)-S7*(C11*ZZ(11)
          1 + S11*C10*C9))
0209       PT( 7, 9,10, 1) = XI1*ZZ(29) + ETA1*(C7*ZZ(31)-S7*C10*S6*S9) + ZET
          1A1*(-C7*C10*S6*S9-S7*ZZ(31))
0210       PT( 7, 9,11, 1) = XI1*(C11*ZZ(13) + S11*C10*S9) + ETA1*C7*(C11*C10
          1*S9-S11*ZZ(13))-ZETA1*S7*(C11*C10*S9-S11*ZZ(13))
0211       PT( 7,10,10, 1) = XI1*ZZ(33) + ETA1*(C7*ZZ(35)-S7*S6*C9*S10) + ZET
          1A1*(-C7*S6*C9*S10-S7*ZZ(35))
0212       PT( 7,10,11, 1) = XI1*ZZ(19) + ETA1*C7*ZZ(18)-ZETA1*S7*ZZ(18)
0213       PT( 7,11,11, 1) = XI1*(-C11*C10*C9-S11*ZZ(12)) + ETA1*C7*(-C11*ZZ(
          112) + S11*C10*C9)-ZETA1*S7*(-C11*ZZ(12) + S11*C10*C9)
0214       PT( 7, 6, 6, 2) = -XI1*S11*C10*C6 + ETA1*(-C7*C11*C10*C6 + S7*C10*
          1S6) + ZETA1*(C7*C10*S6 + S7*C11*C10*C6)
0215       PT( 7, 6, 7, 2) = ETA1*(-C7*C10*C6 + S7*C11*C10*S6) + ZETA1*(C7*C1
          11*C10*S6 + S7*C10*C6)
0216       PT( 7, 6,10, 2) = XI1*S11*S6*S10 + ETA1*(C7*C11*S6*S10 + S7*C6*S10
          1) + ZETA1*(C7*C6*S10-S7*C11*S6*S10)
0217       PT( 7, 6,11, 2) = -XI1*C11*C10*S6 + ETA1*C7*S11*C10*S6-ZETA1*S7*S1
          11*C10*S6
0218       PT( 7, 7, 7, 2) = ETA1*(-C7*ZZ(25) + S7*C10*S6) + ZETA1*(C7*C10*S6
          1 + S7*ZZ(25))
0219       PT( 7, 7,10, 2) = ETA1*(C7*S6*S10-S7*ZZ(23)) + ZETA1*(-C7*ZZ(23)-S
          17*S6*S10)
0220       PT( 7, 7,11, 2) = -ETA1*S7*ZZ(27)-ZETA1*C7*ZZ(27)
0221       PT( 7,10,10, 2) = XI1*ZZ(27) + ETA1*(C7*ZZ(26) + S7*C10*S6) + ZETA
          11*(C7*C10*S6-S7*ZZ(26))
0222       PT( 7,10,11, 2) = XI1*ZZ(23) + ETA1*C7*ZZ(22)-ZETA1*S7*ZZ(22)
0223       PT( 7,11,11, 2) = XI1*ZZ(27) + ETA1*C7*ZZ(26)-ZETA1*S7*ZZ(26)
0224       PT( 7,11,11, 2) = XI1*ZZ(27) + ETA1*C7*ZZ(26)-ZETA1*S7*ZZ(26)
0225       PT( 7, 6, 6, 3) = XI1*S11*ZZ(14) + ETA1*(C7*C11*ZZ(14) + S7*ZZ(16)
          1) + ZETA1*(C7*ZZ(16)-S7*C11*ZZ(14))
0226       PT( 7, 6, 7, 3) = ETA1*(C7*ZZ(14)-S7*C11*ZZ(15)) + ZETA1*(-C7*C11*
          1ZZ(15)-S7*ZZ(14))
0227       PT( 7, 6, 9, 3) = XI1*S11*ZZ(10) + ETA1*(C7*C11*ZZ(10) + S7*ZZ(12)
          1) + ZETA1*(C7*ZZ(12)-S7*C11*ZZ(10))
0228       PT( 7, 6,10, 3) = -XI1*S11*C10*S6*S9 + ETA1*(-C7*C11*C10*S6*S9-S7*
          1C10*C6*S9) + ZETA1*(-C7*C10*C6*S9 + S7*C11*C10*S6*S9)
0229       PT( 7, 6,11, 3) = XI1*C11*ZZ(15)-ETA1*C7*S11*ZZ(15) + ZETA1*S7*S11
          1*ZZ(15)
0230       PT( 7, 7, 7, 3) = ETA1*(-C7*(C11*ZZ(13) + S11*C10*S9)-S7*ZZ(15)) +
          1 ZETA1*(-C7*ZZ(15) + S7*(C11*ZZ(13) + S11*C10*S9))
0231       PT( 7, 7, 9, 3) = ETA1*(C7*ZZ(10)-S7*(C11*ZZ(11) + S11*C10*C9)) +

          1ZETA1*(-C7*(C11*ZZ(11) + S11*C10*C9)-S7*ZZ(10))
0232       PT( 7, 7,10, 3) = ETA1*(-C7*C10*S6*S9-S7*ZZ(31)) + ZETA1*(-C7*ZZ(3
          11) + S7*C10*S6*S9)
0233       PT( 7, 7,11, 3) = -ETA1*S7*(C11*C10*S9-S11*ZZ(13))-ZETA1*C7*(C11*C
          110*S9-S11*ZZ(13))
0234       PT( 7, 9, 9, 3) = XI1*(C11*C10*S9 + S11*ZZ(14)) + ETA1*(C7*(C11*ZZ
          1(14)-S11*C10*S9) + S7*ZZ(16)) + ZETA1*(C7*ZZ(16)-S7*(C11*ZZ(14)-S1
          11*C10*S9))
0235       PT( 7, 9,10, 3) = XI1*ZZ(18) + ETA1*(C7*ZZ(20)-S7*C10*S6*C9) + ZET
          1A1*(-C7*C10*S6*C9-S7*ZZ(20))
0236       PT( 7, 9,11, 3) = XI1*(C11*ZZ(11) + S11*C10*C9) + ETA1*C7*(C11*C10
          1*C9-S11*ZZ(11))-ZETA1*S7*(C11*C10*C9-S11*ZZ(11))
0237       PT( 7,10,10, 3) = XI1*ZZ(37) + ETA1*(C7*ZZ(39) + S7*S6*S9*S10) + Z
          1ETA1*(C7*S6*S9*S10-S7*ZZ(39))
0238       PT( 7,10,11, 3) = XI1*ZZ(31) + ETA1*C7*ZZ(30)-ZETA1*S7*ZZ(30)
0239       PT( 7,11,11, 3) = XI1*(C11*C10*S9-S11*ZZ(13)) + ETA1*C7*(-C11*ZZ(1
          13)-S11*C10*S9)-ZETA1*S7*(-C11*ZZ(13)-S11*C10*S9)
0240       PT( 8, 1,12, 1) = C7*(C11*ZZ(12)-S11*C10*C9) + S7*ZZ(9)
0241       PT( 8, 6,12, 1) = XI*S11*ZZ(9) + Q(1)*(C7*C11*ZZ(9) + S7*ZZ(11)) +
          1 ZETA*(C7*ZZ(11)-S7*C11*ZZ(9))
0242       PT( 8, 7,12, 1) = Q(1)*(C7*ZZ(9)-S7*(C11*ZZ(12)-S11*C10*C9)) + ZET
          1A*(-C7*(C11*ZZ(12)-S11*C10*C9)-S7*ZZ(9))
0243       PT( 8, 9,12, 1) = XI*(-C11*C10*S9 + S11*ZZ(13)) + Q(1)*(C7*(C11*ZZ
          1(13) + S11*C10*S9) + S7*ZZ(15)) + ZETA*(C7*ZZ(15)-S7*(C11*ZZ(13) +
          1 S11*C10*S9))
0244       PT( 8,10,12, 1) = XI*ZZ(17) + Q(1)*(C7*ZZ(19) + S7*C10*S6*C9) + ZE
          1TA*(C7*C10*S6*C9-S7*ZZ(19))
0245       PT( 8,11,12, 1) = XI*(C11*ZZ(12)-S11*C10*C9) + Q(1)*C7*(-C11*C10*C
          19-S11*ZZ(12))-ZETA*S7*(-C11*C10*C9-S11*ZZ(12))
0246       PT( 8, 1,12, 2) = C7*ZZ(25)-S7*C10*S6
0247       PT( 8, 6,12, 2) = -XI*S11*C10*S6 + Q(1)*(-C7*C11*C10*S6-S7*C10*C6)
          1 + ZETA*(-C7*C10*C6 + S7*C11*C10*S6)
0248       PT( 8, 7,12, 2) = Q(1)*(-C7*C10*S6-S7*ZZ(25)) + ZETA*(-C7*ZZ(25) +
          1 S7*C10*S6)
0249       PT( 8,10,12, 2) = XI*ZZ(21) + Q(1)*(C7*ZZ(23) + S7*S6*S10) + ZETA*
          1(C7*S6*S10-S7*ZZ(23))
0250       PT( 8,11,12, 2) = XI*ZZ(25) + Q(1)*C7*ZZ(27)-ZETA*S7*ZZ(27)
0251       PT( 8, 1,12, 3) = C7*(C11*ZZ(13) + S11*C10*S9) + S7*ZZ(15)
0252       PT( 8, 6,12, 3) = XI*S11*ZZ(15) + Q(1)*(C7*C11*ZZ(15) + S7*ZZ(14))
          1 + ZETA*(C7*ZZ(14)-S7*C11*ZZ(15))
0253       PT( 8, 7,12, 3) = Q(1)*(C7*ZZ(15)-S7*(C11*ZZ(13) + S11*C10*S9)) +
          1ZETA*(-C7*(C11*ZZ(13) + S11*C10*S9)-S7*ZZ(15))
0254       PT( 8, 9,12, 3) = XI*(-C11*C10*C9 + S11*ZZ(11)) + Q(1)*(C7*(C11*ZZ
          1(11) + S11*C10*C9) + S7*ZZ(10)) + ZETA*(C7*ZZ(10)-S7*(C11*ZZ(11) +
          1 S11*C10*C9))
0255       PT( 8,10,12, 3) = XI*ZZ(29) + Q(1)*(C7*ZZ(31)-S7*C10*S6*S9) + ZETA
          1*(-C7*C10*S6*S9-S7*ZZ(31))
0256       PT( 8,11,12, 3) = XI*(C11*ZZ(13) + S11*C10*S9) + Q(1)*C7*(C11*C10*
          1S9-S11*ZZ(13))-ZETA*S7*(C11*C10*S9-S11*ZZ(13))
0257       PT( 8, 1, 6, 1) = C7*C11*ZZ(9) + S7*ZZ(11)
0258       PT( 8, 1, 7, 1) = C7*ZZ(9)-S7*(C11*ZZ(12)-S11*C10*C9)
0259       PT( 8, 1, 9, 1) = C7*(C11*ZZ(13) + S11*C10*S9) + S7*ZZ(15)
0260       PT( 8, 1,10, 1) = C7*ZZ(19) + S7*C10*S6*C9
```

```
      PT( 8, 1,11, 1) = C7*(-C11*C10*C9-S11*ZZ(12))
      PT( 8, 6, 6, 1) = XI*S11*ZZ(11) + Q(1)*(C7*C11*ZZ(11) + S7*ZZ(10))
     1 + ZETA*(C7*ZZ(10)-S7*C11*ZZ(11))
      PT( 8, 6, 7, 1) = Q(1)*(C7*ZZ(11)-S7*C11*ZZ(9)) + ZETA*(-C7*C11*ZZ
     1(9)-S7*ZZ(11))
      PT( 8, 6, 9, 1) = XI*S11*ZZ(15) + Q(1)*(C7*C11*ZZ(15) + S7*ZZ(14))
     1 + ZETA*(C7*ZZ(14)-S7*C11*ZZ(15))
      PT( 8, 6,10, 1) = XI*S11*C10*S6*C9 + Q(1)*(C7*C11*C10*S6*C9 + S7*C
     110*C6*C9) + ZETA*(C7*C10*C6*C9-S7*C11*C10*S6*C9)
      PT( 8, 6,11, 1) = XI*C11*ZZ(9)-Q(1)*C7*S11*ZZ(9) + ZETA*S7*S11*ZZ(
     19)
      PT( 8, 7, 7, 1) = Q(1)*(-C7*(C11*ZZ(12)-S11*C10*C9)-S7*ZZ(9)) + ZE
     1TA*(-C7*ZZ(9) + S7*(C11*ZZ(12)-S11*C10*C9))
      PT( 8, 7, 9, 1) = Q(1)*(C7*ZZ(15)-S7*(C11*ZZ(13) + S11*C10*S9)) +
     1ZETA*(-C7*(C11*ZZ(13) + S11*C10*S9)-S7*ZZ(15))
      PT( 8, 7,10, 1) = Q(1)*(C7*C10*S6*C9-S7*ZZ(19)) + ZETA*(-C7*ZZ(19)
     1-S7*C10*S6*C9)
      PT( 8, 7,11, 1) = -Q(1)*S7*(-C11*C10*C9-S11*ZZ(12))-ZETA*C7*(-C11*
     1C10*C9-S11*ZZ(12))
      PT( 8, 9, 9, 1) = XI*(-C11*C10*C9 + S11*ZZ(11)) + Q(1)*(C7*(C11*ZZ
     1(11) + S11*C10*C9) + S7*ZZ(10)) + ZETA*(C7*ZZ(10)-S7*(C11*ZZ(11) +
     1 S11*C10*C9))
      PT( 8, 9,10, 1) = XI*ZZ(29) + Q(1)*(C7*ZZ(31)-S7*C10*S6*S9) + ZETA
     1*(-C7*C10*S6*S9-S7*ZZ(31))
      PT( 8, 9,11, 1) = XI*(C11*ZZ(13) + S11*C10*S9) + Q(1)*C7*(C11*C10*
     1S9-S11*ZZ(13))-ZETA*S7*(C11*C10*S9-S11*ZZ(13))
      PT( 8,10,10, 1) = XI*ZZ(33) + Q(1)*(C7*ZZ(35)-S7*S6*C9*S10) + ZETA
      PT( 8,10,10, 1) = XI*ZZ(33) + Q(1)*(C7*ZZ(35)-S7*S6*C9*S10) + ZETA
     1*(-C7*S6*C9*S10-S7*ZZ(35))
      PT( 8,10,11, 1) = XI*ZZ(19) + Q(1)*C7*ZZ(18)-ZETA*S7*ZZ(18)
      PT( 8,11,11, 1) = XI*(-C11*C10*C9-S11*ZZ(12)) + Q(1)*C7*(-C11*ZZ(1
     12) + S11*C10*C9)-ZETA*S7*(-C11*ZZ(12) + S11*C10*C9)
      PT( 8, 1, 6, 2) = -C7*C11*C10*S6-S7*C10*C6
      PT( 8, 1, 7, 2) = -C7*C10*S6-S7*ZZ(25)
      PT( 8, 1,10, 2) = C7*ZZ(23) + S7*S6*S10
      PT( 8, 1,11, 2) = C7*ZZ(27)
      PT( 8, 6, 6, 2) = -XI*S11*C10*C6 + Q(1)*(-C7*C11*C10*C6 + S7*C10*S
     16) + ZETA*(C7*C10*S6 + S7*C11*C10*C6)
      PT( 8, 6, 7, 2) = Q(1)*(-C7*C10*C6 + S7*C11*C10*S6) + ZETA*(C7*C11
     1*C10*S6 + S7*C10*C6)
      PT( 8, 6,10, 2) = XI*S11*S6*S10 + Q(1)*(C7*C11*S6*S10 + S7*C6*S10)
     1 + ZETA*(C7*C6*S10-S7*C11*S6*S10)
      PT( 8, 6,11, 2) = -XI*C11*C10*S6 + Q(1)*C7*S11*C10*S6-ZETA*S7*S11*
     1C10*S6
      PT( 8, 7, 7, 2) = Q(1)*(-C7*ZZ(25) + S7*C10*S6) + ZETA*(C7*C10*S6
     1+ S7*ZZ(25))
      PT( 8, 7,10, 2) = Q(1)*(C7*S6*S10-S7*ZZ(23)) + ZETA*(-C7*ZZ(23)-S7
     1*S6*S10)
      PT( 8, 7,11, 2) = -Q(1)*S7*ZZ(27)-ZETA*C7*ZZ(27)
      PT( 8,10,10, 2) = XI*ZZ(27) + Q(1)*(C7*ZZ(26) + S7*C10*S6) + ZETA*
     1(C7*C10*S6-S7*ZZ(26))
      PT( 8,10,11, 2) = XI*ZZ(23) + Q(1)*C7*ZZ(22)-ZETA*S7*ZZ(22)
      PT( 8,11,11, 2) = XI*ZZ(27) + Q(1)*C7*ZZ(26)-ZETA*S7*ZZ(26)

      PT( 8, 1, 6, 3) = C7*C11*ZZ(15) + S7*ZZ(14)
      PT( 8, 1, 7, 3) = C7*ZZ(15)-S7*(C11*ZZ(13) + S11*C10*S9)
      PT( 8, 1, 9, 3) = C7*(C11*ZZ(11) + S11*C10*C9) + S7*ZZ(10)
      PT( 8, 1,10, 3) = C7*ZZ(31)-S7*C10*S6*S9
      PT( 8, 1,11, 3) = C7*(C11*C10*S9-S11*ZZ(13))
      PT( 8, 6, 6, 3) = XI*S11*ZZ(14) + Q(1)*(C7*C11*ZZ(14) + S7*ZZ(16))
     1 + ZETA*(C7*ZZ(16)-S7*C11*ZZ(14))
      PT( 8, 6, 7, 3) = Q(1)*(C7*ZZ(14)-S7*C11*ZZ(15)) + ZETA*(-C7*C11*Z
     1Z(15)-S7*ZZ(14))
      PT( 8, 6, 9, 3) = XI*S11*ZZ(10) + Q(1)*(C7*C11*ZZ(10) + S7*ZZ(12))
     1 + ZETA*(C7*ZZ(12)-S7*C11*ZZ(10))
      PT( 8, 6,10, 3) = -XI*S11*C10*S6*S9 + Q(1)*(-C7*C11*C10*S6*S9-S7*C
     110*C6*S9) + ZETA*(-C7*C10*C6*S9 + S7*C11*C10*S6*S9)
      PT( 8, 6,11, 3) = XI*C11*ZZ(15)-Q(1)*C7*S11*ZZ(15) + ZETA*S7*S11*Z
     1Z(15)
      PT( 8, 7, 7, 3) = Q(1)*(-C7*(C11*ZZ(13) + S11*C10*S9)-S7*ZZ(15)) +
     1 ZETA*(-C7*ZZ(15) + S7*(C11*ZZ(13) + S11*C10*S9))
      PT( 8, 7, 9, 3) = Q(1)*(C7*ZZ(10)-S7*(C11*ZZ(11) + S11*C10*C9)) +
     1ZETA*(-C7*(C11*ZZ(11) + S11*C10*C9)-S7*ZZ(10))
      PT( 8, 7,10, 3) = Q(1)*(-C7*C10*S6*S9-S7*ZZ(31)) + ZETA*(-C7*ZZ(31
     1) + S7*C10*S6*S9)
      PT( 8, 7,11, 3) = -Q(1)*S7*(C11*C10*S9-S11*ZZ(13))-ZETA*C7*(C11*C1
     10*S9-S11*ZZ(13))
      PT( 8, 9, 9, 3) = XI*(C11*C10*S9 + S11*ZZ(14)) + Q(1)*(C7*(C11*ZZ(
     114) - S11*C10*S9) + S7*ZZ(16)) + ZETA*(C7*ZZ(16)-S7*(C11*ZZ(14)-S11*
     1C10*S9))
      PT( 8, 9,10, 3) = XI*ZZ(18) + Q(1)*(C7*ZZ(20)-S7*C10*S6*C9) + ZETA
     1*(-C7*C10*S6*C9-S7*ZZ(20))
      PT( 8, 9,11, 3) = XI*(C11*ZZ(11) + S11*C10*C9) + Q(1)*C7*(C11*C10*
     1C9-S11*ZZ(11))-ZETA*S7*(C11*C10*C9-S11*ZZ(11))
      PT( 8,10,10, 3) = XI*ZZ(37) + Q(1)*(C7*ZZ(39) + S7*S6*S9*S10) + ZE
     1TA*(-C7*S6*S9*S10-S7*ZZ(39))
      PT( 8,10,11, 3) = XI*ZZ(31) + Q(1)*C7*ZZ(30)-ZETA*S7*ZZ(30)
      PT( 8,11,11, 3) = XI*(C11*C10*S9-S11*ZZ(13)) + Q(1)*C7*(-C11*ZZ(13
     1)-S11*C10*S9)-ZETA*S7*(-C11*ZZ(13)-S11*C10*S9)
      RETURN
      END
```

```
      SUBROUTINE DER2
C
C     THIS SUBROUTINE DEFINES THE PARTIAL DERIVATIVES PERTAINING TO THE
C     ANGULAR VELOCITIES, POTENTIAL ENERGY, DISSIPATIVE ENERGY, AND
C     GENERALIZED FORCES.
C
      COMMON /DERIV1/Q(11),QD(11),QDD(11)
      COMMON /DERIV2/A1,A2,A3,ASTAR,XKY1,XKY2
      COMMON /DERIV3/A1SUB,A2SUB,A3SUB,XKK1,XKK2,A1BAR
      COMMON /DERIV4/XL(2,2),XM(2,2),XN(2,2),XK(2,2)
      COMMON /DERIV5/E1,E2,E3,FF1,FF2,FF3,D1,D2,D3,XI1,ETA1,ZETA1
      COMMON /DERIV6/B1,B2,B3,B1BAR,XI,ZETA
      COMMON /DERIV7/PG(6,12,12,3)
      COMMON /DERIV8/XIB,EB,ZETAB,XIR,ER,ZETAR,XIC,EC,ZETAC
      COMMON /DERIV9/BLFT,CDFT,RDFT,FOFG
      COMMON /NAME1/PT(8,11,12,3),PW(5,12,12,3),PU(4,11,12,4)
     1 ,PD(3,11,12,4)
      COMMON /NAME7/DA(1,1,1,4),OP(1,1,1,4)
      COMMON /XZZ90/ZZ(90)
      COMMON /DERCON/     G1,G2,G3,D1P,D2P,D3P,C1P,C2P,C3P,ALPHA1,
     1 ALPHA2,ALPHA3,HH1,HH2,HH3
      COMMON /TRIG/C6,S6,C7,S7,C8,S8,C9,S9,C10,S10,C11,S11
C
C     DERIVATIVE FUNCTIONS W/R TO K.E. (ANGULAR TERMS ONLY)
C
      PW( 1,10,12, 1) = QD(9)*C10
      PW( 1, 6,12, 2) = -QD(9)*C10*S6 + QD(10)*C6
      PW( 1,10,12, 2) = -QD(9)*S10*C6
      PW( 1, 6,12, 3) = -QD(9)*C10*C6-QD(10)*S6
      PW( 1,10,12, 3) = QD(9)*S10*S6
      PW( 2,10,12, 1) = QD(9)*C10
      PW( 2, 8,12, 2) = -QD(9)*S8*C10 + QD(10)*C8
      PW( 2,10,12, 2) = -QD(9)*C8*S10
      PW( 2, 8,12, 3) = -QD(9)*C8*C10-QD(10)*S8
      PW( 2,10,12, 3) = QD(9)*S8*S10
      PW( 3, 6,12, 1) = S11*(-QD(9)*C10*S6 + QD(10)*C6)
      PW( 3,10,12, 1) = -QD(9)*S11*S10*C6 + QD(9)*C10*C11
      PW( 3,11,12, 1) = -S11*(QD(6) + QD(9)*S10) + C11*(QD(9)*C10*C6 + Q
     1D(10)*S6)
      PW( 3, 6,12, 2) = C11*(-QD(9)*C10*S6 + QD(10)*C6)
      PW( 3,10,12, 2) = -QD(9)*S11*C10-QD(9)*S10*C11*C6
      PW( 3,11,12, 2) = -S11*(QD(9)*C10*C6 + QD(10)*S6)-C11*(QD(6) + QD(
     19)*S10)
      PW( 3, 6,12, 3) = -QD(9)*C10*C6-QD(10)*S6
      PW( 3,10,12, 3) = QD(9)*S10*S6
      PW( 4, 6,12, 1) = S11*(-QD(9)*C10*S6 + QD(10)*C6)
      PW( 4,10,12, 1) = -QD(9)*S11*S10*C6 + QD(9)*C10*C11
      PW( 4,11,12, 1) = -S11*(QD(6) + QD(9)*S10) + C11*(QD(9)*C10*C6 + Q
     1D(10)*S6)
      PW( 4, 6,12, 2) = C11*C7*(-QD(9)*C10*S6 + QD(10)*C6) + S7*(-QD(9)*
     1C10*C6-QD(10)*S6)
      PW( 4, 7,12, 2) = C7*(QD(11)-QD(9)*C10*S6 + QD(10)*C6)-S7*(-S11*(Q
     1D(6) + QD(9)*S10) + C11*(QD(9)*C10*C6 + QD(10)*S6))
      PW( 4,10,12, 2) = QD(9)*S10*S6*S7 + C7*(-QD(9)*S11*C10-QD(9)*S10*C
     111*C6)
      PW( 4,11,12, 2) = C7*(-S11*(QD(9)*C10*C6 + QD(10)*S6)-C11*(QD(6) +
     1 QD(9)*S10))
      PW( 4, 6,12, 3) = -C11*S7*(-QD(9)*C10*S6 + QD(10)*C6) + C7*(-QD(9)
     1*C10*C6-QD(10)*S6)
      PW( 4, 7,12, 3) = -C7*(-S11*(QD(6) + QD(9)*S10) + C11*(QD(9)*C10*C
     16 + QD(10)*S6))-S7*(QD(11)-QD(9)*C10*S6 + QD(10)*C6)
      PW( 4,10,12, 3) = QD(9)*S10*S6*C7-S7*(-QD(9)*S11*C10-QD(9)*S10*C11
     1*C6)
      PW( 4,11,12, 3) = -S7*(-S11*(QD(9)*C10*C6 + QD(10)*S6)-C11*(QD(6)
     1+ QD(9)*S10))
      PW( 1,12,12, 1) = QD(6) + QD(9)*S10
      PW( 1,12,12, 2) = QD(9)*C10*C6 + QD(10)*S6
      PW( 1,12,12, 3) = -QD(9)*C10*S6 + QD(10)*C6
      PW( 2,12,12, 1) = QD(8) + QD(9)*S10
      PW( 2,12,12, 2) = QD(9)*C8*C10 + QD(10)*S8
      PW( 2,12,12, 3) = -QD(9)*S8*C10 + QD(10)*C8
      PW( 3,12,12, 1) = S11*(QD(9)*C10*C6 + QD(10)*S6) + C11*(QD(6) + QD
     1(9)*S10)
      PW( 3,12,12, 2) = -S11*(QD(6) + QD(9)*S10) + C11*(QD(9)*C10*C6 + Q
     1D(10)*S6)
      PW( 3,12,12, 3) = QD(11)-QD(9)*C10*S6 + QD(10)*C6
      PW( 4,12,12, 1) = QD(7) + S11*(QD(9)*C10*C6 + QD(10)*S6) + C11*(QD
     1(6) + QD(9)*S10)
      PW( 4,12,12, 2) = C7*(-S11*(QD(6) + QD(9)*S10) + C11*(QD(9)*C10*C6
     1 + QD(10)*S6)) + S7*(QD(11)-QD(9)*C10*S6 + QD(10)*C6)
      PW( 4,12,12, 3) = C7*(QD(11)-QD(9)*C10*S6 + QD(10)*C6)-S7*(-S11*(Q
     1D(6) + QD(9)*S10) + C11*(QD(9)*C10*C6 + QD(10)*S6))
      PW( 1,12, 6, 1) = 1
      PW( 1,12, 9, 1) = S10
      PW( 1,12, 9, 2) = C10*C6
      PW( 1,12, 9, 3) = -C10*S6
      PW( 1,12,10, 2) = S6
      PW( 1,12,10, 3) = C6
      PW( 2,12, 8, 1) = 1
      PW( 2,12, 9, 1) = S10
      PW( 2,12, 9, 2) = C8*C10
      PW( 2,12, 9, 3) = -S8*C10
      PW( 2,12,10, 2) = S8
      PW( 2,12,10, 3) = C8
      PW( 3,12, 6, 1) = C11
      PW( 3,12, 6, 2) = -S11
      PW( 3,12, 9, 1) = S11*C10*C6 + S10*C11
      PW( 3,12, 9, 2) = -S11*S10 + C10*C11*C6
      PW( 3,12, 9, 3) = -C10*S6
      PW( 3,12,10, 1) = S11*S6
      PW( 3,12,10, 2) = C11*S6
      PW( 3,12,10, 3) = C6
      PW( 3,12,11, 3) = 1
      PW( 4,12, 6, 1) = C11
      PW( 4,12, 6, 2) = -S11*C7
      PW( 4,12, 6, 3) = S11*S7
```

```
      PW( 4,12, 7, 1) = 1
      PW( 4,12, 9, 1) = S11*C10*C6 + S10*C11
      PW( 4,12, 9, 2) = -C10*S6*S7 + C7*(-S11*S10 + C10*C11*C6)
      PW( 4,12, 9, 3) = -C10*S6*C7-S7*(-S11*S10 + C10*C11*C6)
      PW( 4,12,10, 1) = S11*S6
      PW( 4,12,10, 2) = C11*S6*C7 + C6*S7
      PW( 4,12,10, 3) = -C11*S6*S7 + C6*C7
      PW( 4,12,11, 2) = S7
      PW( 4,12,11, 3) = C7
      PW( 1, 6, 9, 2) = -C10*S6
      PW( 1, 6, 9, 3) = -C10*C6
      PW( 1, 6,10, 2) = C6
      PW( 1, 6,10, 3) = -S6
      PW( 1,10, 9, 1) = C10
      PW( 1,10, 9, 2) = -S10*C6
      PW( 1,10, 9, 3) = S10*S6
      PW( 2, 8, 9, 2) = -S8*C10
      PW( 2, 8, 9, 3) = -C8*C10
      PW( 2, 8,10, 2) = C8
      PW( 2, 8,10, 3) = -S8
      PW( 2,10, 9, 1) = C10
      PW( 2,10, 9, 2) = -C8*S10
      PW( 2,10, 9, 3) = S8*S10
      PW( 3, 6, 9, 1) = -S11*C10*S6
      PW( 3, 6, 9, 2) = -C10*C11*S6
      PW( 3, 6, 9, 3) = -C10*C6
      PW( 3, 6,10, 1) = S11*C6
      PW( 3, 6,10, 2) = C11*C6
      PW( 3, 6,10, 3) = -S6
      PW( 3,10, 9, 1) = -S11*S10*C6 + C10*C11
      PW( 3,10, 9, 2) = -S11*C10-S10*C11*C6
      PW( 3,10, 9, 3) = S10*S6
      PW( 3,11, 6, 1) = -S11
      PW( 3,11, 6, 2) = -C11
      PW( 3,11, 9, 1) = -S11*S10 + C10*C11*C6
      PW( 3,11, 9, 2) = -S11*C10*C6-S10*C11
      PW( 3,11,10, 1) = C11*S6
      PW( 3,11,10, 2) = -S11*S6
      PW( 4, 6, 9, 1) = -S11*C10*S6
      PW( 4, 6, 9, 2) = -C10*C11*S6*C7-C10*C6*S7
      PW( 4, 6, 9, 3) = C10*C11*S6*S7-C10*C6*C7
      PW( 4, 6,10, 1) = S11*C6
      PW( 4, 6,10, 2) = C11*C6*C7-S6*S7
      PW( 4, 6,10, 3) = -C11*C6*S7-S6*C7
      PW( 4, 7, 6, 2) = S11*S7
      PW( 4, 7, 6, 3) = S11*C7
      PW( 4, 7, 9, 2) = -C10*S6*C7-S7*(-S11*S10 + C10*C11*C6)
      PW( 4, 7, 9, 3) = C10*S6*S7-C7*(-S11*S10 + C10*C11*C6)
      PW( 4, 7,10, 2) = -C11*S6*S7 + C6*C7
      PW( 4, 7,10, 3) = -C11*S6*C7-C6*S7
      PW( 4, 7,11, 2) = C7
      PW( 4, 7,11, 3) = -S7
      PW( 4,10, 9, 1) = -S11*S10*C6 + C10*C11

      PW( 4,10, 9, 2) = S10*S6*S7 + C7*(-S11*C10-S10*C11*C6)
      PW( 4,10, 9, 3) = S10*S6*C7-S7*(-S11*C10-S10*C11*C6)
      PW( 4,11, 6, 1) = -S11
      PW( 4,11, 6, 2) = -C11*C7
      PW( 4,11, 6, 3) = C11*S7
      PW( 4,11, 9, 1) = -S11*S10 + C10*C11*C6
      PW( 4,11, 9, 2) = C7*(-S11*C10*C6-S10*C11)
      PW( 4,11, 9, 3) = -S7*(-S11*C10*C6-S10*C11)
      PW( 4,11,10, 1) = C11*S6
      PW( 4,11,10, 2) = -S11*S6*C7
      PW( 4,11,10, 3) = S11*S6*S7
C
C     WUX,WUY,WUZ = WTX,WTY,WTZ
C
      DO 2 J=6,12
      DO 2 K=6,12
      DO 2 L=1,3
    2 PW(5,J,K,L)=PW(4,J,K,L)
C
C     DERIVATIVE FUNCTIONS W/R TO POTENTIAL ENERGY   U2
C
      PU( 2, 4,12, 1) = XK(1,1)*ZZ(60)*S9*S10
      PU( 2, 5,12, 1) = XK(1,1)*ZZ(60)*C9
      PU( 2, 6,12, 1) = XK(1,1)*ZZ(60)*(XN(1,1)*ZZ(14) + XM(1,1)*ZZ(15))
      PU( 2, 9,12, 1) = XK(1,1)*ZZ(60)*(Q(4)*C9*S10-Q(5)*S9-A1*C9*C10 +
     1XN(1,1)*ZZ(10)-A3*S9-XL(1,1)*C9*C10 + XM(1,1)*ZZ(11) + A2*C9*S10)
      PU( 2,10,12, 1) = XK(1,1)*ZZ(60)*(Q(4)*S9*C10 + A1*S9*S10-XN(1,1)*
     1S9*C10*S6 + XL(1,1)*S9*S10 + XM(1,1)*S9*C10*C6 + A2*S9*C10)
      PU( 2, 4,12, 2) = XK(1,2)*ZZ(61)*S9*S10
      PU( 2, 5,12, 2) = XK(1,2)*ZZ(61)*C9
      PU( 2, 6,12, 2) = XK(1,2)*ZZ(61)*(XN(1,2)*ZZ(14) + XM(1,2)*ZZ(15))
      PU( 2, 9,12, 2) = XK(1,2)*ZZ(61)*(Q(4)*C9*S10-Q(5)*S9-A1*C9*C10 +
     1XN(1,2)*ZZ(10)-XL(1,2)*C9*C10 + XM(1,2)*ZZ(11)-A3*S9 + A2*C9*S10)
      PU( 2,10,12, 2) = XK(1,2)*ZZ(61)*(Q(4)*S9*C10 + A1*S9*S10-XN(1,2)*
     1S9*C10*S6 + XL(1,2)*S9*S10 + XM(1,2)*S9*C10*C6 + A2*S9*C10)
      PU( 2, 4,12, 3) = XK(2,1)*ZZ(62)*S9*S10
      PU( 2, 5,12, 3) = XK(2,1)*ZZ(62)*C9
      PU( 2, 6,12, 3) = XK(2,1)*ZZ(62)*(XN(2,1)*ZZ(14) + XM(2,1)*ZZ(15))
      PU( 2, 9,12, 3) = XK(2,1)*ZZ(62)*(XN(2,1)*ZZ(10)-XL(2,1)*C9*C10 +
     1XM(2,1)*ZZ(11) + Q(4)*C9*S10-Q(5)*S9-A1*C9*C10-A3*S9 + A2*C9*S10)
      PU( 2,10,12, 3) = XK(2,1)*ZZ(62)*(-XN(2,1)*S9*C10*S6 + XL(2,1)*S9*
     1S10 + XM(2,1)*S9*C10*C6 + Q(4)*S9*C10 + A1*S9*S10 + A2*S9*C10)
      PU( 2, 4,12, 4) = XK(2,2)*ZZ(63)*S9*S10
      PU( 2, 5,12, 4) = XK(2,2)*ZZ(63)*C9
      PU( 2, 6,12, 4) = XK(2,2)*ZZ(63)*(XN(2,2)*ZZ(14) + XM(2,2)*ZZ(15))
      PU( 2, 9,12, 4) = XK(2,2)*ZZ(63)*(XN(2,2)*ZZ(10) + XM(2,2)*ZZ(11)-
     1XL(2,2)*C9*C10 + Q(4)*C9*S10-Q(5)*S9-A1*C9*C10-A3*S9 + A2*C9*S10)
      PU( 2,10,12, 4) = XK(2,2)*ZZ(63)*(-XN(2,2)*S9*C10*S6 + XM(2,2)*S9*
     1C10*C6 + XL(2,2)*S9*S10 + Q(4)*S9*C10 + A1*S9*S10 + A2*S9*C10)
C
C     DERIVATIVE FUNCTIONS W/R TO POTENTIAL ENERGY U3  PART ONE
C
      PU( 3, 2,12, 1) = -ZZ(53)*(-ZZ(45)*C6*2 + ZZ(46)*S8*2)*.5
```

```
0170      PU( 3, 4,12, 1) = -ZZ(53)*(ZZ(45)*C8*2-ZZ(46)*S8*2)*.5
0171      PU( 3, 5,12, 1) = -ZZ(53)*(ZZ(45)*S8*2 + ZZ(46)*C8*2)*.5
0172      PU( 3, 6,12, 1) = -ZZ(53)*(ZZ(45)*(ZZ(41)*S8 - C8*ZZ(42))*
         12.+ ZZ(46)*(ZZ(41)*C8+S8*ZZ(42))*2.)*.5
0173      PU( 3, 8,12, 1) = -ZZ(53)*(ZZ(45)*(ZZ(43)*C8-ZZ(44)*S8)*2 + ZZ(46)
         1*(-ZZ(43)*S8-ZZ(44)*C8)*2)*.5
     C
     C    DERIVATIVE FUNCTIONS W/R TO POTENTIAL ENERGY  U3 PART TWO
     C
0174      PU( 3, 2,12, 2) = -ZZ(59)*(-ZZ(45)*C8*2 + ZZ(46)*S8*2)*.5
0175      PU( 3, 4,12, 2) = -ZZ(59)*(ZZ(45)*C8*2-ZZ(46)*S8*2)*.5
0176      PU( 3, 5,12, 2) = -ZZ(59)*(ZZ(45)*S8*2 + ZZ(46)*C8*2)*.5
0177      PU( 3, 6,12, 2) = -ZZ(59)*(ZZ(45)*(ZZ(41)*S8 - C8*ZZ(42))*
         12.+ ZZ(46)*(ZZ(41)*C8+S8*ZZ(42))*2.)*.5
0178      PU( 3, 8,12, 2) = -ZZ(59)*(ZZ(45)*(ZZ(43)*C8-ZZ(44)*S8)*2 + ZZ(46)
         1*(-ZZ(43)*S8-ZZ(44)*C8)*2)*.5
     C
     C    DERIVATIVE FUNCTIONS W/R TO POTENTIAL ENERGY U4 PART ONE
     C
0179      PU( 4, 2,12, 1) = XKY1*ZZ(64)*(-C8 + 1)
         1 -Q(2)*(1.-C8)-(-(ALPHA1-Q(3))*C9*S10 + ALPHA2*C10 + ALPHA3*S9*S10
         1 -A2-G2-Q(2))*C8-((ALPHA1-Q(3))*S9-ALPHA3*C9-A3-G3)*S8 + ALPHA2
         1 -A2-G2
0180      PU( 4, 3,12, 1) = XKY1*ZZ(64)*(C8*C9*S10-S8*S9)
         1 -Q(2)*C9*S10*C8 + Q(2)*S9*S8
0181      PU( 4, 8,12, 1) = XKY1*ZZ(64)*(C8*(-A3-G3 + ALPHA3*C9 + S9*(ALPHA1
         1-Q(3)))-S8*(-G2-Q(2)-A2 + ALPHA2*C10 + ALPHA3*S9*S10-C9*S10*(ALPHA
         11-Q(3))))
         1+Q(2)*(-(ALPHA1-Q(3))*C9*S10 + ALPHA2*C10 + ALPHA3*S9*S10-A2-G2
         1 -Q(2))*S8-Q(2)*((ALPHA1-Q(3))*S9 + ALPHA3*C9-A3-G3)*C8
0182      PU( 4, 9,12, 1) = XKY1*ZZ(64)*(C8*(ALPHA3*C9*S10 + S9*S10*(ALPHA1-
         1Q(3))) + S8*(-ALPHA3*S9 + C9*(ALPHA1-Q(3))))
         1 -Q(2)*((ALPHA1-Q(3))*S9*S10 + ALPHA3*C9*S10)*C8-Q(2)*((ALPHA1-
         1 Q(3))*C9-ALPHA3*S9)*S8
0183      PU( 4,10,12, 1) = XKY1*ZZ(64)*C8*(-ALPHA2*S10 + ALPHA3*S9*C10-C9*C
         110*(ALPHA1-Q(3)))
         1 -Q(2)*(-(ALPHA1-Q(3))*C9*C10-ALPHA2*S10 + ALPHA3*S9*C10)*C8
     C
     C    DERIVATIVE FUNCTIONS W/R TO POTENTIAL U4 PART TWO
     C
0184      PU( 4, 2,12, 2) = XKY2*ZZ(65)*(-C8 + 1)
         1 -Q(2)*(1.-C8)-(-(-ALPHA1-Q(3))*C9*S10+ALPHA2*C10+ALPHA3*S9*S10
         1 -A2-G2-Q(2))*C8-((-ALPHA1-Q(3))*S9-ALPHA3*C9-A3-G3)*S8 + ALPHA2
         1 -A2-G2
0185      PU( 4, 3,12, 2) = XKY2*ZZ(65)*(C8*C9*S10-S8*S9)
         1 -Q(2)*C9*S10*C8 + Q(2)*S9*S8
0186      PU( 4, 8,12, 2) = XKY2*ZZ(65)*(C8*(-G3-A3 + ALPHA3*C9 + S9*(-ALPHA
         11-Q(3)))-S8*(-G2-Q(2)-A2 + ALPHA2*C10 + ALPHA3*S9*S10-C9*S10*(-ALP
         1HA1-Q(3))))
         1+Q(2)*(-(-ALPHA1-Q(3))*C9*S10 + ALPHA2*C10 + ALPHA3*S9*S10-A2-G2
         1 -Q(2))*S8-Q(2)*((-ALPHA1-Q(3))*S9 + ALPHA3*C9-A3-G3)*C8
0187      PU( 4, 9,12, 2) = XKY2*ZZ(65)*(C8*(ALPHA3*C9*S10 + S9*S10*(-ALPHA1
         1-Q(3))) + S8*(-ALPHA3*S9 + C9*(-ALPHA1-Q(3))))
         1 -Q(2)*((-ALPHA1-Q(3))*S9*S10 + ALPHA3*C9*S10)*C8-Q(2)*((-ALPHA1-
         1 Q(3))*C9-ALPHA3*S9)*S8
0188      PU( 4,10,12, 2) = XKY2*ZZ(65)*C8*(-ALPHA2*S10 + ALPHA3*S9*C10-C9*C
         110*(-ALPHA1-Q(3)))
         1 -Q(2)*(-(-ALPHA1-Q(3))*C9*C10-ALPHA2*S10 + ALPHA3*S9*C10)*C8
     C
     C    DERIVATIVE FUNCTIONS W/R TO DISSIPATIVE ENERGY UBAR1 PART ONE
     C
0189      PD( 1, 4,12, 1) = S9*S10
0190      PD( 1, 5,12, 1) = C9
0191      PD( 1, 6,12, 1) = XN(1,1)*ZZ(14) + XM(1,1)*ZZ(15)
0192      PD( 1, 9,12, 1) = XN(1,1)*ZZ(10)-XL(1,1)*C9*C10 + XM(1,1)*ZZ(11) +
         1 A2*C9*S10-A3*S9 + Q(4)*C9*S10-Q(5)*S9-A1*C9*C10
0193      PD( 1,10,12, 1) = -XN(1,1)*S9*C10*S6 + XL(1,1)*S9*S10 + XM(1,1)*S9
         1*C10*C6 + A2*S9*C10 + Q(4)*S9*C10 + A1*S9*S10
0194      PD( 1, 4,12, 2) = S9*S10
0195      PD( 1, 5,12, 2) = C9
0196      PD( 1, 6,12, 2) = XM(1,2)*ZZ(15) + XN(1,2)*ZZ(14)
0197      PD( 1, 9,12, 2) = -XL(1,2)*C9*C10 + XM(1,2)*ZZ(11) + XN(1,2)*ZZ(10
         1) + A2*C9*S10-A3*S9 + Q(4)*C9*S10-Q(5)*S9-A1*C9*C10
0198      PD( 1,10,12, 2) = XL(1,2)*S9*S10 + XM(1,2)*S9*C10*C6-XN(1,2)*S9*C1
         10*S6 + A2*S9*C10 + Q(4)*S9*C10 + A1*S9*S10
0199      PD( 1, 4,12, 3) = S9*S10
0200      PD( 1, 5,12, 3) = C9
0201      PD( 1, 6,12, 3) = XM(2,1)*ZZ(15) + XN(2,1)*ZZ(14)
0202      PD( 1, 9,12, 3) = -XL(2,1)*C9*C10 + XM(2,1)*ZZ(11) + XN(2,1)*ZZ(10
         1) + A2*C9*S10-A3*S9 + Q(4)*C9*S10-Q(5)*S9-A1*C9*C10
0203      PD( 1,10,12, 3) = XL(2,1)*S9*S10 + XM(2,1)*S9*C10*C6-XN(2,1)*S9*C1
         10*S6 + A2*S9*C10 + Q(4)*S9*C10 + A1*S9*S10
0204      PD( 1, 4,12, 4) = S9*S10
0205      PD( 1, 5,12, 4) = C9
0206      PD( 1, 6,12, 4) = XM(2,2)*ZZ(15) + XN(2,2)*ZZ(14)
0207      PD( 1, 9,12, 4) = -XL(2,2)*C9*C10 + XM(2,2)*ZZ(11) + XN(2,2)*ZZ(10
         1) + A2*C9*S10-A3*S9 + Q(4)*C9*S10-Q(5)*S9-A1*C9*C10
0208      PD( 1,10,12, 4) = XL(2,2)*S9*S10 + XM(2,2)*S9*C10*C6-XN(2,2)*S9*C1
         10*S6 + A2*S9*C10 + Q(4)*S9*C10 + A1*S9*S10
     C
     C    DERIVATIVE FUNCTIONS W/R TO DISSIPATIVE ENERGY UBAR2 PART ONE
     C
0209      PD( 2, 2,12, 1) = ZZ(82)*(-ZZ(77)*C8*2 + ZZ(78)*S8*2)
0210      PD( 2, 4,12, 1) = ZZ(82)*(ZZ(77)*C8*2-ZZ(78)*S8*2)
0211      PD( 2, 5,12, 1) = ZZ(82)*(ZZ(77)*S8*2 + ZZ(78)*C8*2)
0212      PD( 2, 6,12, 1) = ZZ(82)*(ZZ(77)*(ZZ(79)*S8 + ZZ(80)*C8)*2 + ZZ(78
         1)*(ZZ(79)*C8-ZZ(80)*S8)*2)
0213      PD( 2, 8,12, 1) = ZZ(82)*(ZZ(77)*(ZZ(76)*C8-ZZ(75)*S8)*2 + ZZ(78)*
         1(-ZZ(76)*S8-ZZ(75)*C8)*2)
     C
     C    DERIVATIVE FUNCTIONS W/R TO DISSIPATIVE ENERGY UBAR2 PART TWO
     C
0214      PD( 2, 2,12, 2) = ZZ(81)*(-ZZ(77)*C8*2 + ZZ(78)*S8*2)
0215      PD( 2, 4,12, 2) = ZZ(81)*(ZZ(77)*C8*2-ZZ(78)*S8*2)
0216      PU( 2, 5,12, 2) = ZZ(81)*(ZZ(77)*S8*2 + ZZ(78)*C8*2)
0217      PD( 2, 6,12, 2) = ZZ(81)*(ZZ(77)*(ZZ(79)*S8 + ZZ(80)*C8)*2 + ZZ(78
```

```
      11*(ZZ(79)*C8-ZZ(80)*S8)*2)
0218   PD( 2, 8,12, 2) = ZZ(81)*(ZZ(77)*(ZZ(76)*C8-ZZ(75)*S8)*2 + ZZ(78)*
      1(-ZZ(76)*S8-ZZ(75)*C8)*2)
     C
     C DERIVATIVE FUNCTIONS W/R TO DISSIPATIVE ENERGY UBAR3 PART ONE
     C
0219   PD( 3, 2,12, 1) = ZZ(71)*(ZZ(70)*S8*2-ZZ(69)*C8*2)
0220   PD( 3, 4,12, 1) = ZZ(71)*(-ZZ(70)*S8*2 + ZZ(69)*C8*2)
0221   PD( 3, 5,12, 1) = ZZ(71)*(ZZ(70)*C8*2 + ZZ(69)*S8*2)
0222   PD( 3, 6,12, 1) = ZZ(71)*(ZZ(70)*(-ZZ(72)*S8 + ZZ(73)*C8)*2 + ZZ(6
      19)*(ZZ(72)*C8 + ZZ(73)*S8)*2)
0223   PD( 3, 8,12, 1) = ZZ(71)*(ZZ(70)*(-ZZ(67)*S8-ZZ(66)*C8)*2 + ZZ(69)
      1*(ZZ(67)*C8-ZZ(66)*S8)*2)
     C
     C DERIVATIVE FUNCTIONS W/R TO DISSIPATIVE ENERGY UBAR3 PART TWO
     C
0224   PD( 3, 2,12, 2) = ZZ(74)*(ZZ(70)*S8*2-ZZ(69)*C8*2)
0225   PD( 3, 4,12, 2) = ZZ(74)*(-ZZ(70)*S8*2 + ZZ(69)*C8*2)
0226   PD( 3, 5,12, 2) = ZZ(74)*(ZZ(70)*C8*2 + ZZ(69)*S8*2)
0227   PD( 3, 6,12, 2) = ZZ(74)*(ZZ(70)*(-ZZ(72)*S8 + ZZ(73)*C8)*2 + ZZ(6
      19)*(ZZ(72)*C8 + ZZ(73)*S8)*2)
0228   PD( 3, 8,12, 2) = ZZ(74)*(ZZ(70)*(-ZZ(67)*S8-ZZ(66)*C8)*2 + ZZ(69)
      1*(ZZ(67)*C8-ZZ(66)*S8)*2)
     C
     C DERIVATIVE FUNCTIONS W/R TO GENERALIZED FORCES PART OF VECTOR
     C AB,BB,CB
     C
0229   PG( 1, 1,12, 1) = ZZ(85)*C7 + ZZ(9)*S7
0230   PG( 1, 6,12, 1) = ZETAB*(-ZZ(9)*C11*S7 + ZZ(11)*C7) + XIB*ZZ(9)*S1
      11 + (EB + Q(1))*(ZZ(9)*C11*C7 + ZZ(11)*S7)
0231   PG( 1, 7,12, 1) = ZETAB*(-ZZ(85)*C7-ZZ(9)*S7) + (EB + Q(1))*(-ZZ(8
      15)*S7 + ZZ(9)*C7)
0232   PG( 1, 9,12, 1) = ZETAB*(-ZZ(90)*S7 + ZZ(15)*C7) + XIB*(ZZ(13)*S11
      1-S9*C10*C11) + (EB + Q(1))*(ZZ(90)*C7 + ZZ(15)*S7)
0233   PG( 1,10,12, 1) = ZETAB*(-ZZ(86)*S7 + C9*C10*S6*C7) + XIB*(-C9*C10
      1*S11*C6-C9*S10*C11) + (EB + Q(1))*(ZZ(86)*C7 + C9*C10*S6*S7)
0234   PG( 1,11,12, 1) = -ZETAB*S7*(-ZZ(12)*S11-C9*C10*C11) + XIB*ZZ(85)
      1+ C7*(EB + Q(1))*(-ZZ(12)*S11-C9*C10*C11)
0235   PG( 1, 1,12, 2) = ZZ(87)*C7-C10*S6*S7
0236   PG( 1, 6,12, 2) = ZETAB*(C10*C11*S6*S7-C10*C6*C7)-XIB*C10*S11*S6 +
      1  (EB + Q(1))*(-C10*C11*S6*C7-C10*C6*S7)
0237   PG( 1, 7,12, 2) = ZETAB*(-ZZ(87)*C7 + C10*S6*S7) + (EB + Q(1))*(-Z
      1Z(87)*S7-C10*S6*C7)
0238   PG( 1,10,12, 2) = ZETAB*(S10*S6*C7-S7*(-C10*S11-S10*C11*C6)) + XIB
      1*(C10*C11-S10*S11*C6) + (EB + Q(1))*(S10*S6*S7 + C7*(-C10*S11-S10*
      1C11*C6))
0239   PG( 1,11,12, 2) = -ZETAB*ZZ(88)*S7 + XIB*ZZ(87) + ZZ(88)*C7*(EB +
      1Q(1))
0240   PG( 1, 1,12, 3) = ZZ(90)*C7 + ZZ(15)*S7
0241   PG( 1, 6,12, 3) = ZETAB*(ZZ(14)*C7-ZZ(15)*C11*S7) + XIB*ZZ(15)*S11
      1 + (EB + Q(1))*(ZZ(14)*S7 + ZZ(15)*C11*C7)
0242   PG( 1, 7,12, 3) = ZETAB*(-ZZ(90)*C7-ZZ(15)*S7) + (EB + Q(1))*(-ZZ(
      190)*S7 + ZZ(15)*C7)
0243   PG( 1, 9,12, 3) = ZETAB*(ZZ(10)*C7-S7*(ZZ(11)*C11 + C9*C10*S11)) +
      1 XIB*(ZZ(11)*S11-C9*C10*C11) + (EB + Q(1))*(ZZ(10)*S7 + C7*(ZZ(11)
      1*C11 + C9*C10*S11))
0244   PG( 1,10,12, 3) = ZETAB*(-ZZ(89)*S7-S9*C10*S6*C7) + XIB*(S9*C10*S1
      11*C6 + S9*S10*C11) + (EB + Q(1))*(ZZ(89)*C7-S9*C10*S6*S7)
0245   PG( 1,11,12, 3) = -ZETAB*S7*(-ZZ(13)*S11 + S9*C10*C11) + XIB*ZZ(90
      1) + C7*(EB + Q(1))*(-ZZ(13)*S11 + S9*C10*C11)
     C
     C DERIVATIVE FUNCTION W/R TO GENERALIZED FORCES PART OF VECTOR
     C AR,BR,CR
0246   PG( 2, 1,12, 1) = ZZ(85)*C7 + ZZ(9)*S7
0247   PG( 2, 1,12, 2) = ZZ(87)*C7-C10*S6*S7
0248   PG( 2, 1,12, 3) = ZZ(90)*C7 + ZZ(15)*S7
     C
     C DERIVATIVE FUNCTIONS W/R TO GENERALIZED FORCES   PART OF VECTOR
     C AC,BC,CC
     C
0249   PG( 3, 1,12, 1) = ZZ(85)*C7 + ZZ(9)*S7
0250   PG( 3, 1,12, 2) = ZZ(87)*C7-C10*S6*S7
0251   PG( 3, 1,12, 3) = ZZ(90)*C7 + ZZ(15)*S7
     C
     C VECTOR BUFTA, BUFTB, BUFTC IN GENERALIZED FORCES
     C
0252   PG( 4,12,12, 1) = -BUFT*(ZZ(85)*C7 + ZZ(9)*S7)
0253   PG( 4,12,12, 2) = -BUFT*(ZZ(87)*C7-C10*S6*S7)
0254   PG( 4,12,12, 3) = -BUFT*(ZZ(90)*C7 + ZZ(15)*S7)
     C
     C VECTOR ROFTA, ROFTB, ROFTC IN GENERALIZED FORCES
     C
0255   PG( 5,12,12, 1) = ROFT*(ZZ(85)*C7 + ZZ(9)*S7)
0256   PG( 5,12,12, 2) = ROFT*(ZZ(87)*C7-C10*S6*S7)
0257   PG( 5,12,12, 3) = ROFT*(ZZ(90)*C7 + ZZ(15)*S7)
     C
     C VECTOR CUFTA,CUFTB,CUFTC IN GENERALIZED FORCES
     C
0258   PG( 6,12,12, 1) = CUFT*(ZZ(85)*C7 + ZZ(9)*S7)
0259   PG( 6,12,12, 2) = CUFT*(ZZ(87)*C7-C10*S6*S7)
0260   PG( 6,12,12, 3) = CUFT*(ZZ(90)*C7 + ZZ(15)*S7)
     C
     C DELTA IJ IN U2 OF POTENTIAL ENERGY
     C
0261   DA( 1, 1, 1, 1) = -XN(1,1)-A3 + ZZ(83) + XM(1,1)*ZZ(13) + XN(1,1)*
      1ZZ(15)-XL(1,1)*S9*C10
0262   DA( 1, 1, 1, 2) = -XN(1,2)-A3 + ZZ(83) + XM(1,2)*ZZ(13) + XN(1,2)*
      1ZZ(15)-XL(1,2)*S9*C10
0263   DA( 1, 1, 1, 3) = -XN(2,1)-A3 + ZZ(83)-XL(2,1)*S9*C10 + XM(2,1)*ZZ
      1(13) + XN(2,1)*ZZ(15)
0264   DA( 1, 1, 1, 4) = -XN(2,2)-A3 + ZZ(83)-XL(2,2)*S9*C10 + XM(2,2)*ZZ
      1(13) + XN(2,2)*ZZ(15)
     C
     C DELTA IJ PRIME IN GENERALIZED FORCES
     C
0265   DP( 1, 1, 1, 1) = -XL(1,1) + Q(3)-A1 + ZZ(84) + XM(1,1)*ZZ(12) + X
```

```
     1N(1,1)*ZZ(9) + XL(1,1)*C9*C10
      LP( 1, 1, 1, 2) = -XL(1,2) + Q(3)-A1 + ZZ(84) + XL(1,2)*C9*C10 + X
     1M(1,2)*ZZ(12) + XN(1,2)*ZZ(9)
      LP( 1, 1, 1, 3) = -XL(2,1) + Q(3)-A1 + ZZ(84) + XL(2,1)*C9*C10 + X
     1M(2,1)*ZZ(12) + XN(2,1)*ZZ(9)
      LP( 1, 1, 1, 4) = -XL(2,2) + Q(3)-A1 + ZZ(84) + XL(2,2)*C9*C10 + X
     1M(2,2)*ZZ(12) + XN(2,2)*ZZ(9)
      RETURN
      END
```

```
      SUBROUTINE SOLVE
C
C     THIS SUBROUTINE DECOUPLES THE VECTOR DIFFERENTIAL EQUATION
C     A*QDD=B AND SOLVES FOR THE QDD(I) WHERE I=1,2,....,11
C     A MODIFIED SQUARE ROOT METHOD IS USED.  THIS METHOD TAKES ADVANTAGE
C     OF THE SYMMETRY OF THE MASS MATRIX.
C
      DIMENSION S(10,11),XK(11),C(11)
      COMMON /NAME4/A(11,11),F(11)
      COMMON /DERIV1/Q(11),QD(11),X(11)
      S(1,1) = A(1,1)
      S(1,2) = A(1,2)
      S(1,3) = A(1,3)
      S(1,4) = A(1,4)
      S(1,5) = A(1,5)
      S(1,6) = A(1,6)
      S(1,7) = A(1,7)
      S(1,8) = A(1,8)
      S(1,9) = A(1,9)
      S(1,10) = A(1,10)
      S(1,11) = A(1,11)
      C(1) = S(1,1)**(-1 )
      S(2,3) = A(2,3)-C(1)*S(1,2)*S(1,3)
      S(2,4) = A(2,4)-C(1)*S(1,2)*S(1,4)
      S(2,5) = A(2,5)-C(1)*S(1,2)*S(1,5)
      S(2,6) = A(2,6)-C(1)*S(1,2)*S(1,6)
      S(2,7) = A(2,7)-C(1)*S(1,7)*S(1,2)
      S(2,8) = A(2,8)-C(1)*S(1,8)*S(1,2)
      S(2,9) = A(2,9)-C(1)*S(1,9)*S(1,2)
      S(2,10) = A(2,10)-C(1)*S(1,10)*S(1,2)
      S(2,11) = A(2,11)-C(1)*S(1,11)*S(1,2)
      C(2) = (A(2,2)-C(1)*S(1,2)**2)**(-1 )
      S(3,4) = A(3,4)-C(2)*S(2,3)*S(2,4)-C(1)*S(1,3)*S(1,4)
      S(3,5) = A(3,5)-C(2)*S(2,5)*S(2,3)-C(1)*S(1,3)*S(1,5)
      S(3,6) = A(3,6)-C(2)*S(2,6)*S(2,3)-C(1)*S(1,3)*S(1,6)
      S(3,7) = A(3,7)-C(2)*S(2,7)*S(2,3)-C(1)*S(1,7)*S(1,3)
      S(3,8) = A(3,8)-C(2)*S(2,8)*S(2,3)-C(1)*S(1,8)*S(1,3)
      S(3,9) = A(3,9)-C(2)*S(2,9)*S(2,3)-C(1)*S(1,9)*S(1,3)
      S(3,10) = A(3,10)-S(2,10)*C(2)*S(2,3)-C(1)*S(1,10)*S(1,3)
      S(3,11) = A(3,11)-S(2,11)*C(2)*S(2,3)-C(1)*S(1,11)*S(1,3)
      C(3) = (A(3,3)-C(2)*S(2,3)**2-C(1)*S(1,3)**2)**(-1 )
      S(4,5) = A(4,5)-C(3)*S(3,5)*S(3,4)-C(2)*S(2,5)*S(2,4)-C(1)*S(1,4)*
     1S(1,5)
      S(4,6) = A(4,6)-C(3)*S(3,6)*S(3,4)-C(2)*S(2,6)*S(2,4)-C(1)*S(1,4)*
     1S(1,6)
      S(4,7) = A(4,7)-C(3)*S(3,7)*S(3,4)-C(2)*S(2,7)*S(2,4)-C(1)*S(1,7)*
     1S(1,4)
      S(4,8) = A(4,8)-C(3)*S(3,8)*S(3,4)-C(2)*S(2,8)*S(2,4)-C(1)*S(1,8)*
     1S(1,4)
      S(4,9) = A(4,9)-C(3)*S(3,9)*S(3,4)-C(2)*S(2,9)*S(2,4)-C(1)*S(1,9)*
     1S(1,4)
      S(4,10) = A(4,10)-S(3,10)*C(3)*S(3,4)-S(2,10)*C(2)*S(2,4)-C(1)*S(1
     1,10)*S(1,4)
```

```
0042      S(4,11) = A(4,11)-S(3,11)*C(3)*S(3,4)-S(2,11)*C(2)*S(2,4)-C(1)*S(1
         1,11)*S(1,4)
0043      C(4) = (A(4,4)-C(3)*S(3,4)**2-C(2)*S(2,4)**2-C(1)*S(1,4)**2)**(-1
         1)
0044      S(5,6) = A(5,6)-C(4)*S(4,5)*S(4,6)-C(3)*S(3,5)*S(3,6)-C(2)*S(2,5)*
         1S(2,6)-C(1)*S(1,5)*S(1,6)
0045      S(5,7) = A(5,7)-C(4)*S(4,7)*S(4,5)-C(3)*S(3,5)*S(3,7)-C(2)*S(2,5)*
         1S(2,7)-C(1)*S(1,7)*S(1,5)
0046      S(5,8) = A(5,8)-C(4)*S(4,8)*S(4,5)-C(3)*S(3,5)*S(3,8)-C(2)*S(2,5)*
         1S(2,8)-C(1)*S(1,8)*S(1,5)
0047      S(5,9) = A(5,9)-C(4)*S(4,9)*S(4,5)-C(3)*S(3,5)*S(3,9)-C(2)*S(2,5)*
         1S(2,9)-C(1)*S(1,9)*S(1,5)
0048      S(5,10) = A(5,10)-C(4)*S(4,10)*S(4,5)-S(3,10)*C(3)*S(3,5)-S(2,10)*
         1C(2)*S(2,5)-C(1)*S(1,10)*S(1,5)
0049      S(5,11) = A(5,11)-C(4)*S(4,11)*S(4,5)-S(3,11)*C(3)*S(3,5)-S(2,11)*
         1C(2)*S(2,5)-C(1)*S(1,11)*S(1,5)
0050      C(5) = (A(5,5)-C(4)*S(4,5)**2-C(3)*S(3,5)**2-C(2)*S(2,5)**2-C(1)*S
         1(1,5)**2)**(-1 )
0051      S(6,7) = A(6,7)-C(4)*S(4,7)*S(4,6)-S(5,6)*S(5,7)*C(5)-C(3)*S(3,6)*
         1S(3,7)-C(2)*S(2,6)*S(2,7)-C(1)*S(1,7)*S(1,6)
0052      S(6,8) = A(6,8)-C(4)*S(4,8)*S(4,6)-S(5,6)*S(5,8)*C(5)-C(3)*S(3,6)*
         1S(3,8)-C(2)*S(2,6)*S(2,8)-C(1)*S(1,8)*S(1,6)
0053      S(6,9) = A(6,9)-C(4)*S(4,9)*S(4,6)-S(5,6)*S(5,9)*C(5)-C(3)*S(3,6)*
         1S(3,9)-C(2)*S(2,6)*S(2,9)-C(1)*S(1,9)*S(1,6)
0054      S(6,10) = A(6,10)-C(4)*S(4,10)*S(4,6)-S(5,6)*S(5,10)*C(5)-S(3,10)*
         1C(3)*S(3,6)-S(2,10)*C(2)*S(2,6)-C(1)*S(1,10)*S(1,6)
0055      S(6,11) = A(6,11)-C(4)*S(4,11)*S(4,6)-S(5,6)*S(5,11)*C(5)-S(3,11)*
         1C(3)*S(3,6)-S(2,11)*C(2)*S(2,6)-C(1)*S(1,11)*S(1,6)
0056      C(6) = (A(6,6)-C(4)*S(4,6)**2-C(3)*S(3,6)**2-C(2)*S(2,6)**2-C(1)*S
         1(1,6)**2-S(5,6)**2*C(5))**(-1 )
0057      S(7,8) = A(7,8)-C(4)*S(4,7)*S(4,8)-S(5,7)*S(5,8)*C(5)-C(6)*S(6,8)*
         1S(6,7)-C(3)*S(3,7)*S(3,8)-C(2)*S(2,7)*S(2,8)-C(1)*S(1,7)*S(1,8)
0058      S(7,9) = A(7,9)-C(4)*S(4,7)*S(4,9)-S(5,7)*S(5,9)*C(5)-C(6)*S(6,9)*
         1S(6,7)-C(3)*S(3,7)*S(3,9)-C(2)*S(2,7)*S(2,9)-C(1)*S(1,7)*S(1,9)
0059      S(7,10) = A(7,10)-C(4)*S(4,7)*S(4,10)-S(5,7)*S(5,10)*C(5)-C(6)*S(6
         1,10)*S(6,7)-S(3,10)*C(3)*S(3,7)-S(2,10)*C(2)*S(2,7)-C(1)*S(1,7)*S(
         11,10)
0060      S(7,11) = A(7,11)-C(4)*S(4,7)*S(4,11)-S(5,7)*S(5,11)*C(5)-C(6)*S(6
         1,11)*S(6,7)-S(3,11)*C(3)*S(3,7)-S(2,11)*C(2)*S(2,7)-C(1)*S(1,7)*S(
         11,11)
0061      C(7) = (A(7,7)-C(4)*S(4,7)**2-C(6)*S(6,7)**2-C(3)*S(3,7)**2-C(2)*S
         1(2,7)**2-C(1)*S(1,7)**2-S(5,7)**2*C(5))**(-1 )
0062      S(8,9) = A(8,9)-C(4)*S(4,8)*S(4,9)-S(5,8)*S(5,9)*C(5)-C(7)*S(7,8)*
         1S(7,9)-C(6)*S(6,8)*S(6,9)-C(3)*S(3,8)*S(3,9)-C(2)*S(2,8)*S(2,9)-C(
         11)*S(1,8)*S(1,9)
0063      S(8,10) = A(8,10)-C(4)*S(4,8)*S(4,10)-S(5,8)*S(5,10)*C(5)-C(7)*S(7
         1,8)*S(7,10)-C(6)*S(6,8)*S(6,10)-S(3,10)*C(3)*S(3,8)-S(2,10)*C(2)*S
         1(2,8)-C(1)*S(1,8)*S(1,10)
0064      S(8,11) = A(8,11)-C(4)*S(4,8)*S(4,11)-S(5,8)*S(5,11)*C(5)-C(7)*S(7
         1,8)*S(7,11)-C(6)*S(6,8)*S(6,11)-S(3,11)*C(3)*S(3,8)-S(2,11)*C(2)*S
         1(2,8)-C(1)*S(1,8)*S(1,11)
0065      C(8) = (A(8,8)-C(4)*S(4,8)**2-C(7)*S(7,8)**2-C(6)*S(6,8)**2-C(3)*S
         1(3,8)**2-C(2)*S(2,8)**2-C(1)*S(1,8)**2-S(5,8)**2*C(5))**(-1 )

0066      S(9,10) = A(9,10)-C(4)*S(4,9)*S(4,10)-C(8)*S(8,9)*S(8,10)-C(7)*S(7
         1,9)*S(7,10)-C(6)*S(6,9)*S(6,10)-S(5,9)*S(5,10)*C(5)-S(3,10)*C(3)*S
         1(3,9)-S(2,10)*C(2)*S(2,9)-C(1)*S(1,9)*S(1,10)
0067      S(9,11) = A(9,11)-C(4)*S(4,9)*S(4,11)-C(8)*S(8,9)*S(8,11)-C(7)*S(7
         1,9)*S(7,11)-C(6)*S(6,9)*S(6,11)-S(5,9)*S(5,11)*C(5)-S(3,11)*C(3)*S
         1(3,9)-S(2,11)*C(2)*S(2,9)-C(1)*S(1,9)*S(1,11)
0068      C(9) = (A(9,9)-C(4)*S(4,9)**2-C(8)*S(8,9)**2-C(7)*S(7,9)**2-C(6)*S
         1(6,9)**2-C(3)*S(3,9)**2-C(2)*S(2,9)**2-C(1)*S(1,9)**2-S(5,9)**2*C(
         15))**(-1 )
0069      S(10,11) = A(10,11)-C(4)*S(4,10)*S(4,11)-C(8)*S(8,10)*S(8,11)-S(9,
         110)*S(9,11)*C(9)-C(7)*S(7,10)*S(7,11)-C(6)*S(6,10)*S(6,11)-S(5,10)
         1*S(5,11)*C(5)-S(3,10)*S(3,11)*C(3)-S(2,10)*S(2,11)*C(2)-C(1)*S(1,1
         10)*S(1,11)
0070      C(10) = (A(10,10)-C(4)*S(4,10)**2-C(8)*S(8,10)**2-C(7)*S(7,10)**2-
         1C(6)*S(6,10)**2-C(1)*S(1,10)**2-S(9,10)**2*C(9)-S(5,10)**2*C(5)-S(
         13,10)**2*C(3)-S(2,10)**2*C(2))**(-1 )
0071      C(11) = (A(11,11)-C(4)*S(4,11)**2-C(10)*S(10,11)**2-C(8)*S(8,11)**
         12-C(7)*S(7,11)**2-C(6)*S(6,11)**2-C(1)*S(1,11)**2-S(9,11)**2*C(9)-
         1S(5,11)**2*C(5)-S(3,11)**2*C(3)-S(2,11)**2*C(2))**(-1 )
0072      XK(1) = F(1)
0073      XK(2) = F(2)-XK(1)*C(1)*S(1,2)
0074      XK(3) = F(3)-XK(2)*C(2)*S(2,3)-XK(1)*C(1)*S(1,3)
0075      XK(4) = F(4)-XK(2)*C(2)*S(2,4)-XK(3)*C(3)*S(3,4)-XK(1)*C(1)*S(1,4)
0076      XK(5) = F(5)-C(4)*XK(4)*S(4,5)-XK(2)*C(2)*S(2,5)-XK(3)*C(3)*S(3,5)
         1-XK(1)*C(1)*S(1,5)
0077      XK(6) = F(6)-C(4)*XK(4)*S(4,6)-S(5,6)*XK(5)*C(5)-XK(2)*C(2)*S(2,6)
         1-XK(3)*C(3)*S(3,6)-XK(1)*C(1)*S(1,6)
0078      XK(7) = F(7)-C(4)*XK(4)*S(4,7)-S(5,7)*XK(5)*C(5)-XK(6)*C(6)*S(6,7)
         1-XK(2)*C(2)*S(2,7)-XK(3)*C(3)*S(3,7)-XK(1)*C(1)*S(1,7)
0079      XK(8) = F(8)-C(4)*XK(4)*S(4,8)-S(5,8)*XK(5)*C(5)-XK(6)*C(6)*S(6,8)
         1-XK(7)*C(7)*S(7,8)-XK(2)*C(2)*S(2,8)-XK(3)*C(3)*S(3,8)-XK(1)*C(1)*
         1S(1,8)
0080      XK(9) = F(9)-C(4)*XK(4)*S(4,9)-XK(6)*C(6)*S(6,9)-XK(7)*C(7)*S(7,9)
         1-XK(8)*C(8)*S(8,9)-XK(2)*C(2)*S(2,9)-XK(3)*C(3)*S(3,9)-XK(5)*S(5,9
         1)*C(5)-XK(1)*C(1)*S(1,9)
0081      XK(10) = F(10)-C(4)*XK(4)*S(4,10)-XK(6)*C(6)*S(6,10)-XK(7)*C(7)*S(
         17,10)-XK(8)*C(8)*S(8,10)-XK(9)*S(9,10)*C(9)-XK(2)*S(2,10)*C(2)-XK(
         13)*S(3,10)*C(3)-XK(5)*S(5,10)*C(5)-XK(1)*C(1)*S(1,10)
0082      XK(11) = F(11)-C(4)*XK(4)*S(4,11)-XK(6)*C(6)*S(6,11)-XK(7)*C(7)*S(
         17,11)-XK(8)*C(8)*S(8,11)-XK(9)*S(9,11)*C(9)-XK(10)*C(10)*S(10,11)-
         1XK(2)*S(2,11)*C(2)-XK(3)*S(3,11)*C(3)-XK(5)*S(5,11)*C(5)-XK(1)*C(1
         1)*S(1,11)
0083      X(11) = XK(11)*C(11)
0084      X(10) = -X(11)*C(10)*S(10,11) + XK(10)*C(10)
0085      X(9) = -X(11)*S(9,11)*C(9)-X(10)*S(9,10)*C(9) + XK(9)*C(9)
0086      X(8) = -X(11)*C(8)*S(8,11)-X(10)*C(8)*S(8,10)-X(9)*C(8)*S(8,9) + X
         1K(8)*C(8)
0087      X(7) = -X(11)*C(7)*S(7,11)-X(10)*C(7)*S(7,10)-X(9)*C(7)*S(7,9)-X(8
         1)*C(7)*S(7,8) + XK(7)*C(7)
0088      X(6) = -X(11)*C(6)*S(6,11)-X(10)*C(6)*S(6,10)-X(9)*C(6)*S(6,9)-X(8
         1)*C(6)*S(6,8)-X(7)*C(6)*S(6,7) + XK(6)*C(6)
0089      X(5) = -S(5,6)*X(6)*C(5)-S(5,7)*X(7)*C(5)-S(5,8)*X(8)*C(5)-X(11)*S
         1(5,11)*C(5)-X(10)*S(5,10)*C(5)-X(9)*S(5,9)*C(5) + XK(5)*C(5)
```

```
      X(4) = -C(4)*X(5)*S(4,5)-C(4)*X(11)*S(4,11)-C(4)*X(10)*S(4,10)-C(4
     1)*X(9)*S(4,9)-C(4)*X(6)*S(4,8)-C(4)*X(7)*S(4,7)-C(4)*X(6)*S(4,6) +
     1 C(4)*XK(4)
      X(3) = -X(5)*C(3)*S(3,5)-X(4)*C(3)*S(3,4)-X(11)*S(3,11)*C(3)-X(10)
     1*S(3,10)*C(3)-X(9)*C(3)*S(3,9)-X(8)*C(3)*S(3,8)-X(7)*C(3)*S(3,7)-X
     1(6)*C(3)*S(3,6) + XK(3)*C(3)
      X(2) = -X(5)*C(2)*S(2,5)-X(4)*C(2)*S(2,4)-X(3)*C(2)*S(2,3)-X(11)*S
     1(2,11)*C(2)-X(10)*S(2,10)*C(2)-X(9)*C(2)*S(2,9)-X(8)*C(2)*S(2,8)-X
     1(7)*C(2)*S(2,7)-X(6)*C(2)*S(2,6) + XK(2)*C(2)
      X(1) = -X(5)*C(1)*S(1,5)-X(4)*C(1)*S(1,4)-X(3)*C(1)*S(1,3)-X(2)*C(
     11)*S(1,2)-X(11)*C(1)*S(1,11)-X(10)*C(1)*S(1,10)-X(9)*C(1)*S(1,9)-X
     1(8)*C(1)*S(1,8)-X(7)*C(1)*S(1,7)-X(6)*C(1)*S(1,6) + XK(1)*C(1)
      RETURN
      END

      SUBROUTINE KUTTA
C
C     THIS SUBROUTINE INTEGRATES THE ELEVEN SECOND ORDER NONLINEAR
C     DIFFERENTIAL EQUATIONS WHICH DESCRIBE THE MOTION OF THE M110A1.
C     A FOURTH ORDER RUNGE-KUTTA METHOD IS USED.
C
      DIMENSION QSAVE(11),QDSAVE(11),AK(11,4)
      COMMON /DATA1/TIME,TIMEH,TIMEH2,TIMEH8
      COMMON /DERIV1/Q(11),QD(11),QDD(11)
      COMMON /DERIV9/BLFT,CDFT,ROFT,FOFG
      COMMON /KUTTA2/IBOFT,IROFT,IGOFT
      COMMON /KUTTA3/IBPTS,IRPTS,IGPTS
      COMMON /XREAL1/BRCHX(105),BRCHY(105),RODX(105),RODY(105),
     1 GAMMAX(105),GAMMAY(105)
      DO 5 I=1,11
      QSAVE(I)=Q(I)
    5 QDSAVE(I)=QD(I)
      CALL LINEAR(TIME,BRCHX,BRCHY,BOFT,IBPTS,IBOFT)
      CALL LINEAR(TIME,RODX,RODY,ROFT,IRPTS,IROFT)
      IGOFT=1
      CALL LINEAR(Q(7),GAMMAX,GAMMAY,FOFG,IGPTS,IGOFT)
      CALL NAME
      CALL SOLVE
      DO 1 L=1,11
      AK(L,1)=QDD(L)*TIMEH
      Q(L)=QSAVE(L) + TIMEH2*QDSAVE(L) + TIMEH8*AK(L,1)
    1 QD(L)=QDSAVE(L) + AK(L,1)/2.
      TIME=TIME + TIMEH2
      CALL LINEAR(TIME,BRCHX,BRCHY,BOFT,IBPTS,IBOFT)
      CALL LINEAR(TIME,RODX,RODY,ROFT,IRPTS,IROFT)
      IGOFT=1
      CALL LINEAR(Q(7),GAMMAX,GAMMAY,FOFG,IGPTS,IGOFT)
      CALL NAME
      CALL SOLVE
      DO 2 L=1,11
      AK(L,2)=QDD(L)*TIMEH
    2 QD(L)=QDSAVE(L) + AK(L,2)/2.
      IGOFT=1
      CALL LINEAR(Q(7),GAMMAX,GAMMAY,FOFG,IGPTS,IGOFT)
      CALL NAME
      CALL SOLVE
      DO 3 L=1,11
      AK(L,3)=QDD(L)*TIMEH
      Q(L)=QSAVE(L) + TIMEH*QDSAVE(L) + TIMEH2*AK(L,3)
    3 QD(L)=QDSAVE(L) + AK(L,3)
      TIME=TIME + TIMEH2
      CALL LINEAR(TIME,BRCHX,BRCHY,BOFT,IBPTS,IBOFT)
      CALL LINEAR(TIME,RODX,RODY,ROFT,IRPTS,IROFT)
      IGOFT=1
      CALL LINEAR(Q(7),GAMMAX,GAMMAY,FOFG,IGPTS,IGOFT)
      CALL NAME
      CALL SOLVE
      DO 4 L=1,11

      AK(L,4)=QDD(L)*TIMEH
      Q(L)=QSAVE(L)+TIMEH*(QDSAVE(L) + (AK(L,1) + AK(L,2) + AK(L,3))/6.)
    4 QD(L)=QDSAVE(L) + (AK(L,1) + 2.*(AK(L,2) + AK(L,3)) + AK(L,4))/6.
      RETURN
      END
```

```
      SUBROUTINE NAME
C
C     THIS SUBROUTINE COMBINES THE DIFFERENTIAL EXPRESSIONS IN THE
C     PROPER WAY SUCH THAT IT YIELDS THE COEFFICIENTS MATRIX A OF THE
C     ACCELERATION TERMS AND THE RIGHT HAND SIDES OF THE DIFFERENTIAL
C     EQUATIONS.
C
      DIMENSION SUMM(5,3)
      DIMENSION PKE(5,11,12,3)
      COMMON /DATA2/ XNN2,XNN3,C2,D3,BETAE,GKST,ELANG,CU5
      COMMON /DATA3/BETAT
      COMMON /DERIV1/Q(11),QD(11),QDD(11)
      COMMON /DERIV5/E1,E2,E3,FF1,FF2,FF3,D1,D2,D3,XI1,ETA1,ZETA1
      COMMON /DERIV6/B1,B2,B3,B1BAR,XI,ZETA
      COMMON /DERIV7/PG(6,12,12,3)
      COMMON /DERIV8/XIB,EB,ZETAB,XIR,ER,ZETAR,XIC,EC,ZETAC
      COMMON /DERIV9/BLFT,COFT,ROFT,FOFG
      COMMON /NAME1/PT(8,11,12,3),PW(5,12,12,3),PU(4,11,12,4)
     1 ,PD(3,11,12,4)
      COMMON /NAME2/IEQS,IMASS
      COMMON /NAME3/XMASS(5)
      COMMON /NAME4/AAA(11,11),RHS(11)
      COMMON /NAME5/XIXX(5),XIYY(5),XIZZ(5),XIXY(5),XIYZ(5),XIXZ(5)
      COMMON /NAME6/GRAV,CBRCE,BETA
      COMMON /NAME7/DA(1,1,1,4),DP(1,1,1,4)
      COMMON /NAME8/CC11,CC12,CC21,CC22
      COMMON /NAME9/HOCR1
C
C     EVALUATE PARTIAL DERIVATIVES
C
      CALL DERFUC
C
C     DEFINE FIRST AND SECOND DERIVATIVE TERMS FOR THE KINETIC ENERGY
C     EXPRESSIONS FOR THE FIVE MASSES (NOT INCLUDING THE ANGULAR TERMS)
C
      DO 2 J=1,11
      DO 2 K=1,12
      DO 2 L=1,3
      PKE(1,J,K,L)=PT(1,J,K,L) + PT(2,J,K,L)
      PKE(2,J,K,L)=PT(1,J,K,L) + PT(3,J,K,L)
      PKE(3,J,K,L)=PKE(1,J,K,L) + PT(4,J,K,L) + PT(5,J,K,L)
      PKE(4,J,K,L)=PKE(1,J,K,L)+PT(4,J,K,L)+PT(6,J,K,L)+PT(7,J,K,L)
    2 PKE(5,J,K,L)=PKE(1,J,K,L)+PT(4,J,K,L)+PT(6,J,K,L)+PT(8,J,K,L)
C
C     DEFINE COEFFICIENTS MATRIX FOR QDD TERMS RESULTING FROM KINETIC
C     ENERGY (DOES NOT INCLUDE THE ANGULAR TERMS)
C     ONLY THE UPPER TRIANGULAR TERMS ARE DEFINED HERE
C
      DO 5 J=1,IEQS
      DO 5 K=J,IEQS
      SUM=0.
      DO 4 I=1,IMASS
      DO 4 L=1,3

    4 SUM=SUM + XMASS(I)*PKE(I,J,12,L)*PKE(I,K,12,L)
    5 AAA(J,K)=SUM
C
C     INITIALIZE RIGHT HAND SIDES TO ZERO
C
      DO 6 I=1,IEQS
    6 RHS(I)=0.
C
C     KINETIC ENERGY EXPRESSIONS FOR TERM 3 (NOT INCLUDING ANGULAR TERMS)
C
      DO 8 I=1,IMASS
      DO 8 L=1,3
      SUMM(I,L)=0.
      DO 8 K=1,IEQS
      DO 8 J=1,IEQS
    8 SUMM(I,L)=SUMM(I,L) + PKE(I,J,K,L)*QD(J)*QD(K)
      DO 9 J=1,IEQS
      DO 9 I=1,IMASS
      DO 9 L=1,3
    9 RHS(J)=RHS(J) + XMASS(I)*PKE(I,J,12,L)*SUMM(I,L)
C
C     DEFINE THE REST OF THE COEFFICIENTS MATRIX FOR QDD TERMS RESULTING
C     FROM THE ANGULAR VELOCITIES IN THE KINETIC ENERGY EXPRESSIONS.
C
      DO 11 J=1,IEQS
      DO 11 K=J,IEQS
      SUM=0.
      DO 10 I=1,IMASS
      SUM=SUM + (0.5*XIXX(I)*PW(I,12,J,1) - XIXY(I)*PW(I,12,J,2))*
     1 PW(I,12,K,1)
     2 + (0.5*XIYY(I)*PW(I,12,J,2) - XIYZ(I)*PW(I,12,J,3))*
     3 PW(I,12,K,2)
     4 + (0.5*XIZZ(I)*PW(I,12,J,3) - XIXZ(I)*PW(I,12,J,1))*
     5 PW(I,12,K,3)
   10 SUM=SUM + (0.5*XIXX(I)*PW(I,12,K,1) - XIXY(I)*PW(I,12,K,2))*
     1 PW(I,12,J,1)
     2 + (0.5*XIYY(I)*PW(I,12,K,2) - XIYZ(I)*PW(I,12,K,3))*
     3 PW(I,12,J,2)
     4 + (0.5*XIZZ(I)*PW(I,12,K,3) - XIXZ(I)*PW(I,12,K,1))*
     5 PW(I,12,J,3)
   11 AAA(J,K)=AAA(J,K) + SUM
C
C     OBTAIN ENTIRE COEFFICIENTS MATRIX
C
      DO 50 I=1,11
      DO 50 J=1,11
   50 AAA(J,I)=AAA(I,J)
C
C     RIGHT HAND SIDES OF ANGULAR TERMS
C
      DO 13 J=1,IEQS
      SUM=0.
      DO 13 K=1,IEQS
```

```
0062          DO 13 I=1,IMASS
0063          SUM=SUM + (0.5*XIXX(I)*PW(I,K,J,1) - XIXY(I)*PW(I,K,J,2))*
             1 PW(I,12,12,1)*QD(K)
             2 + (0.5*XIYY(I)*PW(I,K,J,2) - XIYZ(I)*PW(I,K,J,3))*
             3 PW(I,12,12,2)*QD(K)
             4 + (0.5*XIZZ(I)*PW(I,K,J,3) - XIXZ(I)*PW(I,K,J,1))*
             5 PW(I,12,12,3)*QD(K)
0064          SUM=SUM + (0.5*XIXX(I)*PW(I,12,J,1) - XIXY(I)*PW(I,12,J,2))*
             1 PW(I,K,12,1)*CD(K)
             2 + (0.5*XIYY(I)*PW(I,12,J,2) - XIYZ(I)*PW(I,12,J,3))*
             3 PW(I,K,12,2)*CD(K)
             4 + (0.5*XIZZ(I)*PW(I,12,J,3) - XIXZ(I)*PW(I,12,J,1))*
             5 PW(I,K,12,3)*CD(K)
0065          SUM=SUM + (0.5*XIXX(I)*PW(I,K,12,1) - XIXY(I)*PW(I,K,12,2))*
             1 PW(I,12,J,1)*CD(K)
             2 + (0.5*XIYY(I)*PW(I,K,12,2) - XIYZ(I)*PW(I,K,12,3))*
             3 PW(I,12,J,2)*CD(K)
             4 + (0.5*XIZZ(I)*PW(I,K,12,3) - XIXZ(I)*PW(I,K,12,1))*
             5 PW(I,12,J,3)*CD(K)
0066          SUM=SUM + (0.5*XIXX(I)*PW(I,12,12,1) - XIXY(I)*PW(I,12,12,2))*
             1 PW(I,K,J,1)*CD(K)
             2 + (0.5*XIYY(I)*PW(I,12,12,2) - XIYZ(I)*PW(I,12,12,3))*
             3 PW(I,K,J,2)*CD(K)
             4 + (0.5*XIZZ(I)*PW(I,12,12,3) - XIXZ(I)*PW(I,12,12,1))*
             5 PW(I,K,J,3)*CD(K)
0067       13 RHS(J)=RHS(J) + SUM
         C
         C    TERMS FOR    MINUS PARTIAL K.E. W/R TO THE GENERALIZED COORDINATES
         C    (ANGULAR TERMS ONLY).
         C
0068          DO 493 J=1,IEQS
0069          SUM=0.
0070          DO 492 I=1,IMASS
0071          SUM=SUM + (0.5*XIXX(I)*PW(I,J,12,1) - XIXY(I)*PW(I,J,12,2))*
             1 PW(I,12,12,1)
             2 + (0.5*XIYY(I)*PW(I,J,12,2) - XIYZ(I)*PW(I,J,12,3))*PW(I,12,12,2)
             3 + (0.5*XIZZ(I)*PW(I,J,12,3) - XIXZ(I)*PW(I,J,12,1))*PW(I,12,12,3)
0072      492 SUM=SUM + (0.5*XIXX(I)*PW(I,12,12,1) - XIXY(I)*PW(I,12,12,2))*
             1 PW(I,J,12,1)
             2 +(0.5*XIYY(I)*PW(I,12,12,2)-XIYZ(I)*PW(I,12,12,3))*PW(I,J,12,2)
             3 + (0.5*XIZZ(I)*PW(I,12,12,3)-XIXZ(I)*PW(I,12,12,1))*PW(I,J,12,3)
0073      493 RHS(J)=RHS(J) - SUM
         C
         C    RIGHT HAND SIDES COMPLETE AT THIS POINT FOR K.E.(TRANSLATION AND ROTATION)
         C
         C
         C    RIGHT HAND SIDES DUE TO POTENTIAL ENERGY OF U1
         C
0074          DO 14 J=1,IEQS
0075       14 RHS(J)=RHS(J) + GRAV*(XMASS(1)*PKE(1,J,12,3) + XMASS(2)*
             1 PKE(2,J,12,3) + XMASS(3)*PKE(3,J,12,3) + XMASS(4)*PKE(4,J,12,3) +
             2 XMASS(5)*PKE(5,J,12,3))
         C
         C    RIGHT HAND SIDES DUE TO POTENTIAL ENERGY OF U2
         C
0076          DO 15 J=1,IEQS
0077       15 RHS(J)=RHS(J)+PU(2,J,12,1)+PU(2,J,12,2)+PU(2,J,12,3)+PU(2,J,12,4)
         C
         C    RIGHT HAND SIDES DUE TO POTENTIAL ENERGY OF U3
         C
0078          DO 16 J=1,IEQS
0079       16 RHS(J)=RHS(J) + PU(3,J,12,1) + PU(3,J,12,2)
         C
         C    RIGHT HAND SIDES DUE TO POTENTIAL ENERGY OF U4
         C
0080          DO 17 J=1,IEQS
0081       17 RHS(J)=RHS(J) + PU(4,J,12,1) + PU(4,J,12,2)
         C
         C    RHS DUE TO POTENTIAL ENERGY OF U5
         C
0082          S7=SIN(Q(7))
0083          C7=COS(Q(7))
0084          XLENGH = D1*D1 + (XNN2-(D2-D2)*C7 - (D3-D3)*S7)**2 + (XNN3 +
             1 (D2-D2)*S7 - (D3-D3)*C7)**2
0085          XLENGH=SQRT(XLENGH)
0086          TERM1 = FOFG*(XNN2*(-XNN3-(D2-D2)*S7 + (D3-D3)*C7) - XNN3*(-XNN2
             1 + (D2-D2)*C7 + (D3-D3)*S7))/XLENGH
0087          GAMST=Q(7) - ELANG
0088          BBETAE=BUFT*ZETAB
0089          IF(BBETAE .GT. BETAE) BBETAE=BETAE
0090          IF(QD(7) .LE. 0.) BBETAE=0.
0091          RHS(7)=RHS(7) +BBETAE             + HOOR1*(GKST*GAMST +
             1 QD(7)*CU5) - TERM1
         C
         C    RHS DUE TO POTENTIAL ENERGY OF U6
         C
0092          BBETAT=BUFT*(-XI)
0093          IF(ABS(BBETAT) .GT. BETAT)  BBETAT=BETAT
0094          IF(QD(11) .LE. 0.) BBETAT=0.
0095          RHS(11)=RHS(11) + BBETAT
         C
         C    RIGHT HAND SIDES DUE TO DISSIPATIVE ENERGY OF UBAR1
         C
0096          SUM1=0.
0097          SUM2=0.
0098          SUM3=0.
0099          SUM4=0.
0100          DO 22 J=4,10
0101          SUM1=SUM1 + PD(1,J,12,1)*CD(J)
0102          SUM2=SUM2 + PD(1,J,12,2)*CD(J)
0103          SUM3=SUM3 + PD(1,J,12,3)*CD(J)
0104       22 SUM4=SUM4 + PD(1,J,12,4)*CD(J)
0105          DO 23 K=4,10
0106       23 RHS(K)=RHS(K) + CC11*SUM1*PD(1,K,12,1) + CC12*SUM2*PD(1,K,12,2) +
             1 CC21*SUM3*PD(1,K,12,3) + CC22*SUM4*PD(1,K,12,4)
         C
```

```
C       RIGHT HAND SIDES DUE TO DISSIPATIVE ENEKGY OF UBAR2
C
        SUM1=0.
        SUM2=0.
        DO 21 J=1,IEQS
        SUM1=SUM1 + PD(2,J,12,1)*CD(J)
     21 SUM2=SUM2 + PD(2,J,12,2)*CD(J)
        DO 20 K=1,IECS
     20 RHS(K)=RHS(K) + CBRCE*(SUM1*PD(2,K,12,1) + SUM2*PD(2,K,12,2))
C
C       RIGHT HAND SIDES DUE TO DISSIPATIVE ENERGY OF UBAR3
C
        SUM1=0.
        SUM2=0.
        DO 18 J=1,11
        SUM1=SUM1 + PD(3,J,12,1)*CD(J)
     18 SUM2=SUM2 + PD(3,J,12,2)*CD(J)
        DO 19 K=1,11
     19 RHS(K)=RHS(K) + BETA*(PD(3,K,12,1)*SUM1**2 + PD(3,K,12,2)*SUM2**2)
C
C       BRING ALL TERMS FROM LEFT SIDE OF EQUATIONS TO RIGHT HAND SIDE
C
        DO 24 I=1,11
     24 RHS(I)=-RHS(I)
C
C       GENERALIZED FORCE DUE TO BREECH FORCE
C
        DO 100 J=1,11
        DO 100 L=1,3
    100 PG(1,J,12,L)=PKE(1,J,12,L)+PT(4,J,12,L)+PT(6,J,12,L)+PG(1,J,12,L)
        DO 102 J=1,IEQS
        SUM=0.
        DO 101 L=1,3
    101 SUM=SUM + PG(4,12,12,L)*PG(1,J,12,L)
    102 RHS(J)=RHS(J) + SUM
C
C       GENERALIZED FORCE DUE TO ROD PULL
C
        DO 103 L=1,3
    103 PG(2,1,12,L)=PG(2,1,12,L)+PKE(1,1,12,L)+PT(4,1,12,L)+PT(6,1,12,L)
        SUM=0.
        DO 104 L=1,3
    104 SUM=SUM + PG(5,12,12,L)*PG(2,1,12,L)
        RHS(1)=RHS(1) + SUM
        RETURN
        END
```

```
        SUBROUTINE LINEAR(A,X,Y,VV,M,I)
        DIMENSION X(105),Y(105)
C
C       THIS SUBROUTINE CBTAINS VALUES BETWEEN ADJACENT ENTRIES BY LINEAR
C       INTERPOLATION.  LAGRANGE'S INTERPOLATION FORMULA IS USED.
C
        IF(I .LE. 0)I=1
      2 IF(A-X(I))3,1,1
      1 I=I+1
        IF(M-I)3,3,2
      3 I=I-1
        VV=Y(I)*(A-X(I+1))/(X(I)-X(I+1))+Y(I+1)*(A-X(I))/(X(I+1)-X(I))
        RETURN
        END
```

```
      BLOCK DATA
      COMMON /DATA1/TIME,TIMEH,TIMEH2,TIMEH8
      COMMON /DATA2/ XNN2,XNN3,C2,D3,BETAE,GKST,ELANG,CUS
      COMMON /DATA3/BETAT
      COMMON /DERIV1/Q(11),QD(11),QDD(11)
      COMMON /DERIV2/A1,A2,A3,ASTAR,XKY1,XKY2
      COMMON /DERIV3/A1SUB,A2SUB,A3SUB,XKK1,XKK2,A1BAR
      COMMON /DERIV4/XL(2,2),XM(2,2),XN(2,2),XK(2,2)
      COMMON /DERIV5/E1,E2,E3,FF1,FF2,FF3,D1,D2,D3,XI1,ETA1,ZETA1
      COMMON /DERIV6/B1,B2,B3,B1BAR,XI,ZETA
      COMMON /DERIV8/XIB,EB,ZETAB,XIR,ER,ZETAR,XIC,EC,ZETAC
      COMMON /DERIV9/BOFT,COFT,RDFT,FOFG
      COMMON /NAME2/IEQS,IMASS
      COMMON /NAME3/XMASS(5)
      COMMON /NAME5/XIXX(5),XIYY(5),XIZZ(5),XIXY(5),XIYZ(5),XIXZ(5)
      COMMON /NAME6/GRAV,CBRCE,BETA
      COMMON /NAME8/CC11,CC12,CC21,CC22
      COMMON /KUTTA2/IBOFT,IROFT,IGOFT
      DATA Q(1),Q(2),Q(3),Q(4),Q(5),Q(6),Q(7),Q(8),Q(9),Q(10),Q(11)
     1 /83.675,0.0,0.0,0.0,-.14,0.0,0.0,0.0,0.0,0.0,0.0/
      DATA QD/11*0.0/
      DATA A1,A2,A3,ASTAR/0.0,19.6,-13.6,51.0/
      DATA A1SUB,A2SUB,A3SUB/39.5,-125.1,-7.15/
      DATA XKK1,XKK2/100000.,100000./
      DATA A1BAR/-39.5/
      DATA XK(1,1),XK(1,2),XK(2,1),XK(2,2)/4*100000./
      DATA XL(1,1),XL(1,2),XL(2,1),XL(2,2)/53.0,53.0,-53.0,-53.0/
      DATA XM(1,1),XM(1,2),XM(2,1),XM(2,2)/48.03,-99.97,48.03,-99.97/
      DATA XN(1,1),XN(1,2),XN(2,1),XN(2,2)/-32.4,-32.4,-32.4,-32.4/
      DATA B1,B2,B3,B1BAR/39.5,-124.6,9.6,-39.5/
      DATA E1,E2,E3/0.0,-84.6,2.0/
      DATA FF1,FF2,FF3/0.0,5.2,16.65/
      DATA D1,D2,D3/0.0,27.375,47.85/
      DATA XI1,ETA1,ZETA1/0.0,23.625,-11.25/
      DATA XI,ZETA/0.0,1.8/
      DATA XKY1,XKY2/2*100000./
      DATA IEQS,IMASS/11,5/
      DATA XMASS(1),XMASS(2),XMASS(3),XMASS(4),XMASS(5)/88.601,
     1 4.663,20.207,10.868,36.308/
      DATA XIB,EB,ZETAB/0.0,-16.0,1.8/
      DATA XIR,ER,ZETAR/-3.938,-14.0,-14.312/
      DATA XIC,EC,ZETAC/3.217,-14.0,-12.062/
      DATA BOFT,COFT,RCFT/3*0.0/
      DATA XIXY/5*0.0/
      DATA XIYZ/5*0.0/
      DATA XIXZ/5*0.0/
      DATA XIXX(1),XIXX(2),XIXX(3),XIXX(4),XIXX(5)
     1 /238342.,1502.6,6476.7,12953.,132124./
      DATA XIYY(1),XIYY(2),XIYY(3),XIYY(4),XIYY(5)
     1 /93782.,6114.,3886.,2279.8,673.6/
      DATA XIZZ(1),XIZZ(2),XIZZ(3),XIZZ(4),XIZZ(5)
     1 /260622.,1502.6,3886.,2279.8,132124./
      DATA GRAV,CBRCE,BETA/386.,0.0,0.0/

      DATA CC11,CC12,CC21,CC22/4*0.0/
      DATA TIME,TIMEH/0.0,0.0005/
      DATA IBOFT,IRUFT,IGOFT/1,1,1/
      DATA XNN2,XNN3,D2,D3,BETAE,GKST,ELANG,CUS
     1 /73.0,-10.5,27.375,65.5, 100000.,3*0./
      DATA BETAT/0.0/
      END
```

TIME

ETA	V	X	Y	Z	PHI	GAM	NU	THET	PSI	TAU
0.0005										
83.6747	-0.0004	-0.0000	-0.0000	-0.1400	-0.0000	0.0000	-0.0000	-0.0000	0.0000	-0.0000
-1.3103	-1.4131	0.0000	-0.0868	-0.0015	-0.0001	0.0001	-0.0155	0.0000	-0.0000	0.0000
-3173.0664	-2814.7271	0.0040	-173.9220	2.1001	-0.1507	-0.0376	-30.7734	0.0005	-0.0000	0.0002
0.0010										
83.6736	-0.0014	0.0000	-0.0001	-0.1400	-0.0000	0.0000	-0.0000	0.0000	-0.0000	0.0000
-3.0575	-2.8090	0.0000	-0.1744	-0.0004	-0.0002	0.0001	-0.0307	0.0000	-0.0000	0.0000
-3815.5720	-2764.5732	0.0031	-176.7103	2.3345	-0.1157	-0.0559	-29.6741	0.0004	-0.0000	0.0001
0.0015										
83.6716	-0.0032	0.0000	-0.0002	-0.1400	-0.0000	0.0000	-0.0000	0.0000	-0.0000	0.0000
-5.0484	-4.1709	0.0000	-0.2637	0.0008	-0.0002	0.0000	-0.0451	0.0000	-0.0000	0.0000
-4147.8633	-2680.6804	0.0013	-180.2143	2.3917	-0.0728	-0.0765	-27.8573	0.0002	-0.0000	0.0001
0.0020										
83.6685	-0.0056	0.0000	-0.0004	-0.1400	-0.0000	0.0000	-0.0001	0.0000	-0.0000	0.0000
-7.1326	-5.4830	0.0000	-0.3549	0.0011	-0.0003	0.0000	-0.0584	0.0000	-0.0000	0.0000
-4193.2461	-2563.6677	0.0043	-186.0071	-7.6169	-0.1948	0.3169	-25.3544	0.0005	-0.0000	0.0002
0.0025										
83.6643	-0.0086	0.0000	-0.0006	-0.1400	-0.0000	0.0000	-0.0001	0.0000	-0.0000	0.0000
-9.8862	-6.7294	0.0000	-0.4487	-0.0003	-0.0003	0.0001	-0.0703	0.0000	-0.0000	0.0000
-6827.7148	-2417.9802	0.0021	-190.8412	-7.8755	-0.1386	0.2923	-22.2401	0.0003	-0.0000	0.0001
0.0030										
83.6583	-0.0123	0.0000	-0.0008	-0.1400	-0.0000	0.0000	-0.0001	0.0000	-0.0000	0.0000
-14.4217	-7.8962	0.0000	-0.5447	0.0007	-0.0003	0.0000	-0.0805	0.0000	-0.0000	0.0000
-11320.4141	-2243.2422	-0.0004	-194.7598	1.6270	0.0962	-0.1506	-18.5563	-0.0000	0.0000	-0.0000
0.0035										
83.6494	-0.0165	0.0000	-0.0011	-0.1400	-0.0000	0.0000	-0.0002	0.0000	-0.0000	0.0000
-21.4621	-8.9698	0.0000	-0.6476	-0.0001	-0.0002	-0.0000	-0.0888	0.0000	-0.0000	0.0000
-16840.9570	-2047.8862	0.0026	-216.8883	-4.7696	0.1973	0.1109	-14.4284	0.0003	-0.0000	0.0001
0.0040										
83.6364	-0.0212	0.0000	-0.0014	-0.1400	-0.0000	0.0000	-0.0002	0.0000	-0.0000	0.0000
-31.3036	-9.9379	0.0000	-0.7610	0.0008	-0.0000	-0.0001	-0.0949	0.0000	-0.0000	0.0000
-22525.3555	-1825.4407	0.0014	-236.5583	8.4888	0.6485	-0.4617	-9.8626	0.0002	-0.0000	0.0001
0.0045										
83.6177	-0.0264	0.0000	-0.0019	-0.1400	-0.0000	0.0000	-0.0003	0.0000	-0.0000	0.0000
-43.7253	-10.7916	0.0000	-0.8854	0.0009	0.0003	-0.0002	-0.0986	0.0000	-0.0000	0.0000
-27154.6914	-1587.4905	0.0026	-259.7427	1.8229	0.7590	-0.2020	-5.0231	0.0003	-0.0000	0.0001
0.0050										
83.5922	-0.0320	0.0000	-0.0023	-0.1400	-0.0000	-0.0000	-0.0003	0.0000	-0.0000	0.0000
-58.5205	-11.5215	0.0000	-1.0208	0.0026	0.0008	-0.0003	-0.0998	0.0000	-0.0000	0.0000
-32026.2969	-1334.2512	0.0017	-281.7900	4.9631	1.0443	-0.3588	0.0175	0.0002	-0.0000	0.0001
0.0055										
83.5588	-0.0379	0.0000	-0.0029	-0.1400	0.0000	-0.0000	-0.0004	0.0000	-0.0000	0.0000
-75.7521	-12.1219	0.0000	-1.1681	0.0060	0.0014	-0.0005	-0.0985	0.0000	-0.0000	0.0000
-36900.1641	-1068.9670	-0.0001	-307.1282	8.7887	1.3725	-0.5404	5.1234	-0.0000	0.0000	-0.0000

AD______________ Accession No.____________

ARRADCOM, Large Claiber Weapons System Laboratory, Weapons Division, Rock Island, IL 61201

A DYNAMIC ANALYSIS OF THE SELF PROPELLED 8 INCH, M110A1 HOWITZER UTILIZING FORMAC

Weapons Laboratory Rep. R-TR-77-025, Sep 76 - Apr 77, 151 pp incl appendixes, (DA Project No. 1W662603AH78, AMS Code No. 6.26.03.A), Unclassified report by Thomas D. Streeter and Robert H. Coberly

This report presents a new method for solving Lagrange's equations of motion utilizing FORMAC. An application using this technique is given with an eleven degree-of-freedom probelm which describes the motion of the M110A1, a self-propelled "1" howitzer under dynamic conditions of firing. A computer program has been written, is operational, and the listing is contained in the appendix. This report is an endeavor to automate the generation and solution of the equations of motion for dynamical systems.

UNCLASSIFIED

1. M110A1 2. FORMAC

3. Dynamic 4. Math Model

5. Simulation 6. FORTRAN

7. Self Propelled 8. LaGrange

DISTRIBUTION:

Distribution Unlimited

AD______________ Accession No.____________

ARRADCOM, Large Claiber Weapons System Laboratory, Weapons Division, Rock Island, IL 61201

A DYNAMIC ANALYSIS OF THE SELF PROPELLED 8 INCH, M110A1 HOWITZER UTILIZING FORMAC

Weapons Laboratory Rep. R-TR-77-025, Sep 76 - Apr 77, 151 pp incl appendixes, (DA Project No. 1W662603AH78, AMS Code No. 6.26.03.A), Unclassified report by Thomas D. Streeter and Robert H. Coberly

This report presents a new method for solving Lagrange's equations of motion utilizing FORMAC. An application using this technique is given with an eleven degree-of-freedom probelm which describes the motion of the M110A1, a self-propelled "1" howitzer under dynamic conditions of firing. A computer program has been written, is operational, and the listing is contained in the appendix. This report is an endeavor to automate the generation and solution of the equations of motion for dynamical systems.

UNCLASSIFIED

1. M110A1 2. FORMAC

3. Dynamic 4. Math Model

5. Simulation 6. FORTRAN

7. Self Propelled 8. LaGrange

DISTRIBUTION:

Distribution Unlimited